12

Technical Report: NAVTRAEQUIPCEN 75-C-0057-1

SIMULATION OF AN AIR CUSHION VEHICLE

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

March 1977

Final Report for Period January 1975 - December 1976

DOD DISTRIBUTION STATEMENT

Approved for public release;
distribution unlimited.

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

18

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

~~1. REPORT NUMBER~~ 19
NAVTRAEQUIPCEN 75-C-0057- 1 ✓

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 SIMULATION OF AN AIR CUSHION VEHICLE.

7 ~~5. TYPE OF REPORT & PERIOD COVERED~~
Final Report.
January 1975 — December 76,

~~6. PERFORMING ORG. REPORT NUMBER~~

~~7. AUTHOR(s)~~
10 John F./Loud

8. CONTRACT OR GRANT NUMBER(s)
15 N61339-75-C-0057 new

9. PERFORMING ORGANIZATION NAME AND ADDRESS
The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
3741-03P~~01B~~

11. CONTROLLING OFFICE NAME AND ADDRESS
Computer Laboratory
Naval Training Equipment Center
Orlando, Florida 32813

11 ~~12. REPORT DATE~~
March 1977

13. NUMBER OF PAGES
280

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
12 249p.

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Air Cushion Vehicle, ACV Simulation, Real-time Simulation, Vehicle Generated Waves, Seaway. Air Flow between compartments, air cushion pressure, predictor-correction integration, JEFF-B, ocean, pilot trainer, equations of motion.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the components, architecture, and operations of the ACV-LC real-time digital simulation. The system configuration is described in detail, including the pilot station controls and displays. The mathematical model of the equations of motion, propulsion dynamics, dynamic air flow of the cushions, seaway generation, and the dynamics of the pressure wave (vehicle generated waves) is completely presented and defined. Descriptions of the model are included for readability.

UNCLASSIFIED LB

408 386

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This report describes the development of the digital computer real-time program for the mathematical model of a JEFFB Amphibious Assault Landing Craft (ACV-LC). The mathematical model is presented in the report NAVTRAEQUIPCEN 73-C-0138-1.

The real-time program is designed to be implemented in the Sigma 7 computer facility of the Naval Training Equipment Center. The system integrating the program and other components is being used as an ACV-LC experimental training device.

In the areas of Vehicle Generated Wave, Offshore Wave, and the compartment pressure models, special program techniques, simplification and remodeling of the mathematical model were made in order to make the simulation run in real-time.

TABLE OF CONTENTS

TABLE OF CONTENTS(Cont'd)

List of Illustrations

List of Illustrations (Cont'd)

List of Tables

SECTION I
INTRODUCTION

This report details an all digital real-time simulation developed for the Naval Training Equipment Center (NAVTRAEQUIPCEN) of the Bell Aerospace JEFF-B design, an air cushioned vehicle landing craft (ACV-LC). This simulation is intended for use as a pilot trainer; therefore, only gross vehicle characteristics and performance can be expected. This report contains a presentation of the model and a description of the implementation. The mathematical model used has been detailed in the report: Mathematical Model of an Air Cushion Vehicle.

A complete description of specific model areas as they differ is included for correlation purposes where possible and necessary. Some of the areas of modelling completely differ from the given; other areas differ only in format or form. In all cases, however, the symbology has been changed but is adequately defined.

This report discusses techniques used in the simulation development. In several cases, special techniques were developed to make the simulation run in real-time.

Complete operational procedures are given in detail in order to provide the operator with all the tools to run the simulation and to make on-line changes as necessary (i.e. initial conditions).

This report is not intended, necessarily, for use in duplicating the simulation. Although, a complete description of all components is provided, no programming techniques and methods, as such have been included.

This all digital simulation was developed to use the equipment and be supported by the personnel at NAVTRAEQUIPCEN's general purpose computer facility. This facility includes a Xerox Sigma 7 Computer; complete hybrid interface of analog to digital converters, digital to analog converters,and discrete inputs; and trunking station with removable patchboard. Also connected to the Sigma 7 is a line drawing CRT with appropriate interface. Necessary for simulation of an ACV-LC is the mockup of the pilot cabin with a set of controls duplicating placement and operating characteristics as are located on the vehicle.

The CRT display is used to provide the visual effects necessary to dock the ACV-LC inside the stern well of a mother ship. This display is external to the cockpit and provides a three dimensional picture for approach and docking maneuvers.

SECTION II
ACV-LC JEFF-B

The BellAerospace Air Cushioned Vehicle Landing Craft, JEFF-B, has been modelled and simulated real-time in a pilot trainer. There are two very important areas in understanding the operation of the simulation and the content of this report. First, the size and shape of the vehicle and type of effectors determine the ship's handling characteristics. Second, the type of controls and displays available to the pilot define the method of driving the ship.

VEHICLE CONFIGURATION

The ACV-LC is virtually rectangular in shape with a flat bottom and a semi-rigid inflatable skirt surrounding the periphery of the hull. This skirt hangs below the hull bottom approximately 4.5 ft. allowing adequate space for an air volume or mass for flotation purposes. Air is pumped into each quarter of this cushion compartment by large fans, providing the pressure necess..y for the lift to keep the hull above water. Leakage of air occurs at the boundary between skirt and water surface and is vented through moveable nozzles at the top of the hull. (See Figure 1).

These 2 thrust nozzles (1 each, port, starboard) are located in the forward part of the ship at the top of the super structure and provide large amounts of thrust. These nozzles are capable of directing this thrust through 360° and move in unison.

Located at the stern of the ACV are 2 huge ducted, variable pitch propellers (1 each, port, starboard) that provide thrust in the longitudinal direction. Each propeller is driven directly from its output power shaft which is governor controlled and coupled to 3 gas turbines. The air inlet fans are geared to the power shaft.

Located behind each propeller are large air vanes (rudders) which move together and provide steering control. When used in conjunction with the bow thrust nozzles, the ship can make coordinated turns, sharp turns, and sloppy turns; also the ship can be moved laterally.

The pilot cabin is located at the bow on the starboard side at the top of the super structure allowing good peripheral vision from the ACV-LC. The mother ship and its stern well can be seen clearly for docking purposes.

The air pressure under the hull tends to flatten the water surface and to create a depression when hovering. This effect causes water to be pushed outward from beneath the ship causing vehicle generated waves (VGW). These waves are of little consequence to the vehicle performance except for the bow wave, if moving forward. However, the pressure distribution is greatly effected. This bow wave is pushed by the flat non V hull and causes the humping speed phenomenon. This bow wave causes sufficient drag that once created at constant low speeds it is not possible to accelerate the vehicle through it. This pressure wave has a maximum height in the velocity range 16-20 knots. At slowly increasing forward speed, there is a region of decreased acceleration; then, as the vehicle climbs over the wave, a sudden increase in acceleration occurs as the vehicle rides down the wave.

However, if the vehicle's trajectory is such that it crosses its own path, then interference of the past VGW with the present VGW does occur, impacting the vehicle maneuverability.

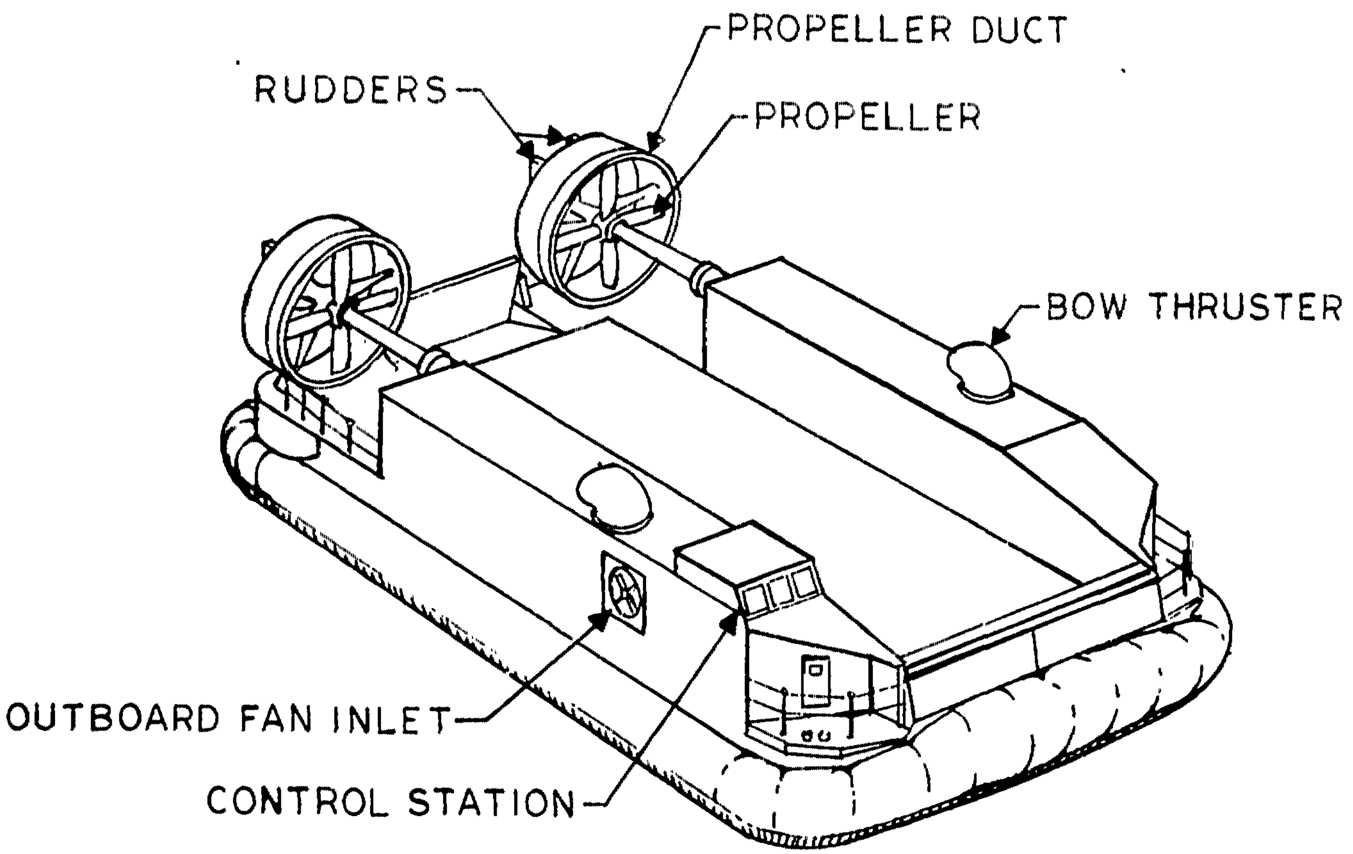


FIGURE 1. ACV-LC JEFF-B Effector Configuration

PILOT CONTROLS AND DISPLAYS

The pilot needs all controls to maneuver the vehicle. Since the vehicle has no automated control system, the individual control devices for all effectors must be available to the pilot. (See Table 1). All 6 gas turbine throttles (3 PORT, 3 STBD) command turbine speed. The STBD and PORT governor controls set the output power shaft speed according to the command and the related gas turbine speeds. The power shafts govern the propeller speeds (propulsion), fan speeds (compartment air flows), and, consequently, the thrust nozzle air flows. There are 3 propeller pitch control devices, one each PORT and STBD to individually set each propeller pitch, and a vernier to change both propeller pitch commands simultaneously by moving the steering wheel fwd/aft. The thrust nozzles are directed by the steering wheel and by a switch to direct the thrust fwd/aft. The rudders are directed by foot pedal commands.

For forward motion, the pilot commands equally the 6 gas turbines and the 2 power shafts. To command the correct forward thrust for a given speed the pilot directs the nozzle fwd or aft and sets the two propeller pitch commands equally to the proper angle. The pilot can then vary the propeller pitch command by tilting the steering wheel to adjust the vehicle's desired forward speed.

To execute a turning maneuver, the pilot can command several of the effectors in a variety of ways. Differential commands may be set into the power shaft governors and/or propeller pitch. The steering wheel may be turned to deflect the nozzle thrust and/or the rudder pedals may be used to deflect the propeller air flow. Normally, a turn would be executed by coordinated commands to the nozzle thrust and rudder deflections. A tighter turn could be made by also commanding the propellers pitch differentially.

Visual feedback is provided to the pilot so that he will know the state of the effectors and the vehicle. Gas turbine and power shaft speeds are examples of effector state. Forward speed and heading are examples of vehicle state. See Table 2 for the complete list of ACV-LC displays.

A pilot station configuration of controls and displays is shown in Figure 2.

TABLE 1. ACV-LC PILOT CONTROLS

STBD gas turbine 1, 2, 3 throttles
PORT gas turbine 1, 2, 3 throttles
STBD power shaft governor
PORT power shaft governor
STBD propeller pitch
PORT propeller pitch
Vernier propeller pitch
Thrust nozzle angle
Thrust nozzle switch
Rudder pedals

TABLE 2. ACV-LC PILOT DISPLAYS

Forward velocity
Lateral velocity
Heading
Pitch angle
Roll angle
Apparent wind speed
Apparent wind angle
Hull height at CG
STBD propeller pitch
PORT propeller pitch
STBD rudder angle
PORT rudder angle
Thrust nozzle angle
STBD power shaft speed
PORT power shaft speed
STBD gas turbine speeds (3)
PORT gas turbine speeds (3)
STBD gas turbine fuel flow rates (3)
PORT gas turbine fuel flow rates (3)
STBD gas turbine torques (3)
PORT gas turbine torques (3)

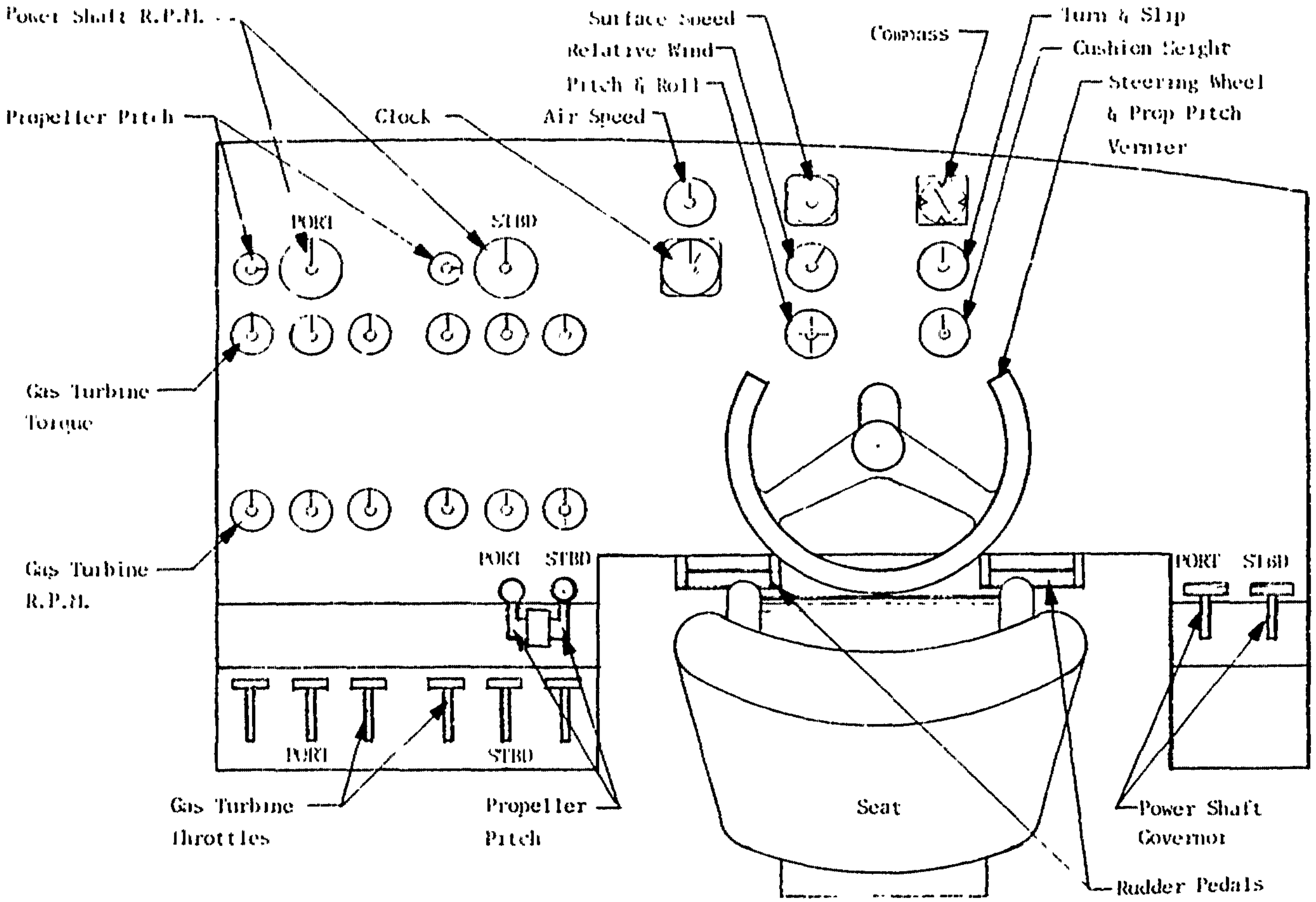


Figure 2. Pilot station controls and displays

SECTION III

MATHEMATICAL MODEL

The complete mathematical model, as used in the simulation, is presented here for completeness. Most equations differ little from the given model. Symbology is different and the programming does follow the symbology as outlined in this section. Solution techniques are described where special programming techniques are needed to solve equations. No other programming techniques are described. In general, however, one programming technique, modularity, is employed throughout the simulation program. A separate module or subroutine is developed for each separate topic.

EQUATIONS OF MOTION

This section describes the equations of motion and the particular force and moment components due to each effector or environmental effect as acting upon a point mass. These individual force or moment components are then summed to yield the total acting about each axis centered at the pilot.

FORMAT. The standard form of the equations of motion for the ACV-LC are shown. Manipulations of these equations were performed in order to eliminate acceleration cross-coupling terms. This produced a set of nonsimultaneous equations in 6 variables which could be solved routinely.

The equations of motion show the summation of forces and moments about each axis. This summation has been broken down into the various components and performed in a separate subroutine.

Subroutine: Motion

Equations of Motion

$$\dot{u} = rv-qw + X_{CG}(q^2+r^2)- Y_{CG}(pq-\dot{r}) -Z_{CG}(pr +\dot{q}) + \frac{\Sigma F_x}{m}$$

$$\dot{v} = pw-ru + Y_{CG}(r^2+p^2) -Z_{CG}(qr-\dot{p}) -X_{CG}(qp+\dot{r}) + \frac{\Sigma F_y}{m}$$

$$\dot{w} = qu-pr+Z_{CG}(p^2+q^2) -X_{CG}(rp-\dot{q}) -Y_{CG}(rq+\dot{p}) + \frac{\Sigma F_z}{m}$$

$$\dot{p} = -qr \frac{I_z-I_y}{I_x} - \frac{m}{I_x} \Big[Y_{CG}(\dot{w}+pv-qu) -Z_{CG}(\dot{v} +ru-pw) + X_{CG}Y_{CG}(pr-\dot{q})$$
$$-X_{CG}Z_{CG}(pq+\dot{r}) +Y_{CG}Z_{CG}(r^2-q^2)\Big] + \frac{\Sigma M_x}{I_x}$$

$$\dot{q} = -rp \frac{I_x-I_z}{I_y} - \frac{m}{I_y} \Big[Z_{CG}(\dot{u} + qw-rv) - X_{CG}(\dot{w} + pv-qu) + Y_{CG}Z_{CG}(qp -\dot{r})$$
$$-Y_{CG}X_{CG}(qr+\dot{p}) +Z_{CG}X_{CG}(p^2-r^2)\Big] + \frac{\Sigma M_y}{I_y}$$

$$\dot{r} = -pq \frac{I_y - I_x}{I_z} - \frac{m}{I_z} \Big[X_{CG}(\dot{v} + ru-pw) -Y_{CG}(\dot{u} + qw-rv) + Z_{CG}X_{CG}(rq-\dot{p})$$
$$-Z_{CG}Y_{CG}(rp +\dot{q}) +X_{CG}Y_{CG}(q^2-p^2)\Big] + \frac{\Sigma M_z}{I_z}$$

$$\dot{u} = rv - qw - Y_{CG}pq - Z_{CG}\,pr + X_{CG}\,(q^2 + r^2) - Z_{CG}\,\dot{q} + Y_{CG}\,\dot{r} + \frac{1}{m}\left[\Sigma F_x\right]$$

$$\dot{v} = pw - ru - Z_{CG}\,qr - X_{CG}\,qp + Y_{CG}(r^2 + p^2) - X_{CG}\dot{r} + Z_{CG}\,\dot{p} + \frac{1}{m}\left[\Sigma F_y\right]$$

$$\dot{w} = qu - pv - X_{CG}rp - Y_{CG}\,rq + Z_{CG}\,(p^2 + q^2) - Y_{CG}\,\dot{p} + X_{CG}\,\dot{q} + \frac{1}{m}\left[\Sigma F_z\right]$$

$$\dot{p} = \frac{I_y - I_z}{I_x}\,qr - \frac{m}{I_x}\left[Y_{CG}(pv-qu) - Z_{CG}(ru-pw) + X_{CG}Y_{CG}pr - X_{CG}Z_{CG}pq + Y_{CG}Z_{CG}(r^2 - q^2)\right] - \frac{m}{I_x}\left[-Z_{CG}\dot{v} + Y_{CG}\dot{w} - X_{CG}Y_{CG}\dot{q} - X_{CG}Z_{CG}\dot{r}\right] + \frac{1}{I_x}\left[\Sigma M_x\right]$$

$$\dot{q} = \frac{I_z - I_x}{I_y}\,rp - \frac{m}{I_y}\left[Z_{CG}(qw-rv) - X_{CG}(pv-qu) + Y_{CG}Z_{CG}qp - X_{CG}Y_{CG}qr + Z_{CG}X_{CG}(p^2 - r^2)\right] - \frac{m}{I_y}\left[-X_{CG}\dot{w} + Z_{CG}\dot{u} - Y_{CG}Z_{CG}\dot{r} - X_{CG}Y_{CG}\dot{p}\right] + \frac{1}{I_y}\left[\Sigma M_y\right]$$

$$\dot{r} = \frac{I_x - I_y}{I_z}\,pq - \frac{m}{I_z}\left[X_{CG}(ru-pw) - Y_{CG}(qw-rv) + Z_{CG}X_{CG}rq - Z_{CG}Y_{CG}rp + X_{CG}Y_{CG}(q^2 - p^2)\right] - \frac{m}{I_z}\left[-Y_{CG}\dot{u} + X_{CG}\dot{v} - Z_{CG}X_{CG}\dot{p} - Z_{CG}Y_{CG}\dot{q}\right] + \frac{1}{I_z}\left[\Sigma M_z\right]$$

$$G_x = rv - qw - Y_{CG}\,pq - Z_{CG}\,pr + X_{CG}\,(q^2 + r^2) + \frac{1}{m}\left[\Sigma F_x\right]$$

$$G_y = pw - ru - Z_{CG}\,qr - X_{CG}\,qp + Y_{CG}\,(r^2 + p^2) + \frac{1}{m}\left[\Sigma F_y\right]$$

$$G_z = qu - pv - X_{CG}\,rp - Y_{CG}\,rq + Z_{CG}\,(p^2 + q^2) + \frac{1}{m}\left[\Sigma F_z\right]$$

$$G_K = \frac{I_y - I_z}{I_x}\,qr - \frac{m}{I_x}\Big[Y_{CG}(pv-qu) - Z_{CG}\,(ru-pw) + X_{CG}Y_{CG}\,pr - X_{CG}Z_{CG}\,pq$$
$$+ Y_{CG}Z_{CG}\,(r^2-q^2)\Big] + \frac{1}{I_x}\left[\Sigma M_x\right]$$

$$G_M = \frac{I_z - I_x}{I_y}\,rp - \frac{m}{I_y}\Big[Z_{CG}\,(qw-rv) - X_{CG}\,(pv-qu) + Y_{CG}\,Z_{CG}\,qp - Y_{CG}X_{CG}\,qr$$
$$+ Z_{CG}X_{CG}\,(p^2-r^2)\Big] + \frac{1}{I_y}\left[\Sigma M_y\right]$$

$$G_N = \frac{I_x - I_y}{I_z}\,pq - \frac{m}{I_z}\Big[X_{CG}\,(ru-pw) - Y_{CG}(qw-rv) + Z_{CG}X_{CG}\,rq - Z_{CC}Y_{CG}\,rp$$
$$+ X_{CG}\;Y_{CG}\,(q^2-p^2)\Big] + \frac{1}{I_z}\left[\Sigma M_z\right]$$

$$\dot{u} = G_x - Z_{CG}\,\dot{q} + Y_{CG}\,\dot{r}$$

$$\dot{v} = G_y - X_{CG}\,\dot{r} + Z_{CG}\,\dot{p}$$

$$\dot{w} = G_z - Y_{CG}\,\dot{p} + X_{CG}\,\dot{q}$$

$$\dot{p} = G_K - \frac{m}{I_x}\left[-Z_{CG}\,\dot{v} + Y_{CG}\,\dot{w} - X_{CG}\,Y_{CG}\,\dot{q} - X_{CG} Z_{CG}\,\dot{r}\right]$$

$$\dot{q} = G_M - \frac{m}{I_y}\left[-X_{CG}\,\dot{w} + Z_{CG}\dot{u} - Y_{CG}\,Z_{CG}\,\dot{r} - Y_{CG}\,X_{CG}\,\dot{p}\right]$$

$$\dot{r} = G_N - \frac{m}{I_z}\left[-Y_{CG}\,\dot{u} + X_{CG}\,\dot{v} - Z_{CG}\,X_{CG}\,\dot{p} - Z_{CG}\,Y_{CG}\,\dot{q}\right]$$

$$\dot{p} = \left[G_K + \frac{m}{I_x}\,(Z_{CG}\;\;G_Y - Y_{CG}\;\;G_Z)\right] \Big/ \left[1.0 - \frac{m}{I_x}\;(Z^2_{CG} + Y^2_{CG})\right]$$

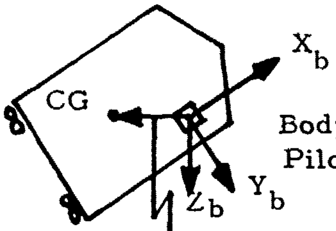
$$\dot{q} = \left[G_M + \frac{m}{I_y}\,(X_{CG}\;\;G_Z - Z_{CG}\;\;G_X)\right] \Big/ \left[1.0 - \frac{m}{I_y}\;(X^2_{CG} + Z^2_{CG})\right]$$

$$\dot{r} = \left[G_N + \frac{m}{I_z}\,(Y_{CG}\;\;G_X - X_{CG}\;\;G_Y)\right] \Big/ \left[1.0 - \frac{m}{I_z}\;(Y^2_{CG} + X^2_{CG})\right]$$

$$\dot{u} = G_X + Y_{CG}\;\;\dot{r} - Z_{CG}\;\;\dot{q}$$

$$\dot{v} = G_Y + Z_{CG}\;\;\dot{p} - X_{CG}\;\;\dot{r}$$

$$\dot{w} = G_Z + X_{CG}\qquad \cdot Y_{CG}\;\;\dot{p}$$



Body Coordinates Centered at Pilot Cabin

$(X_{CG} Y_{CG}, Z_{CG})_b$ = Center of Gravity

$\dot{u}$	Acceleration (Linear) in direction X_b (Ft/Sec^2)
$\dot{v}$	Acceleration (Linear) in direction Y_b (Ft/Sec^2)
$\dot{w}$	Acceleration (Linear) in direction Z_b (Ft/Sec^2)
$\dot{p}$	Acceleration (Angular) about axis X_b (Rad/Sec^2)
$\dot{q}$	Acceleration (Angular) about axis Y_b (Rad/Sec^2)
$\dot{r}$	Acceleration (Angular) about axis Z_b (Rad/Sec^2)
u	Velocity (Linear) in direction X_b (Ft/Sec)
v	Velocity (Linear) in direction Y_b (Ft/Sec)
w	Velocity (Linear) in direction Z_b (Ft/Sec)
p	Velocity (Angular) about axis X_b (Rad/Sec)
q	Velocity (Angular) about axis Y_b (Rad/Sec)
r	Velocity (Angular) about axis Z_b (Rad/Sec)
X_{CG}	Position of CG measured along X_b = -30.0 Ft.
Y_{CG}	Position of CG measured along Y_b = -18.0 Ft.
Z_{CG}	Position of CG measured along Z_b = + 8.0 Ft.
m	ACV MASS = 10879.5 Slugs
I_x	Moment of Inertia about Axis X_b =5.672 x 10^6 SLUG-FT^2
I_y	Moment of Inertia about Axis Y_b =1.629 x 10^7 SLUG-FT^2
I_z	Moment of Inertia about Axis Z_b =2.057 x 10^7 SLUG-FT^2
ΣF_x	Summation of Forces Acting on Vehicle in Direction X_b (Lbs)
ΣF_y	Summation of Forces Acting on Vehicle in Direction Y_b (Lbs)
ΣF_z	Summation of Forces Acting on Vehicle in Direction Z_b (Lbs)
ΣM_x	Summation of Moments Acting on Vehicle about Axis X_b (Lb-Ft)
ΣM_y	Summation of Moments Acting on Vehicle about Axis Y_b (Lb-Ft)
ΣM_z	Summation of Moments Acting on Vehicle about Axis Z_b (Lb-Ft)

Subroutine: Forces

$$\Sigma F_x = X_{PROP} + X_{NOZ} + X_{RUDDER} + X_{AERO} + X_{AIR} + X_{GRAV} + X_{SKIRT} + X_{SEA}$$

$$\Sigma F_y = Y_{NOZ} + Y_{RUDDER} + Y_{AERO} + Y_{AIR} + Y_{GRAV} + Y_{SKIRT} + Y_{DUCT} + Y_{SEA}$$

$$\Sigma F_z = Z_{GRAV} + Z_{CUSH}$$

$$\Sigma M_x = K_{NOZ} + K_{AERO} + K_{AIR} + K_{GRAV} + K_{DUCT} + K_{CUSH} + K_{SKIRT}$$

$$\Sigma M_y = M_{NOZ} + M_{AERO} + M_{AIR} + M_{GRAV} + M_{SKIRT} + M_{CUSH}$$

$$\Sigma M_z = N_{PROP} + N_{NOZ} + N_{RUDDER} + N_{AERO} + N_{AIR} + N_{GRAV} + N_{SKIRT} + N_{DUCT} + N_{DAMP} + N_{SEA}$$

$\Sigma F_x, \Sigma F_y, \Sigma F_z, \Sigma M_x, \Sigma M_y, \Sigma M_z$

Subscripts	Forces & Moments due to:
PROP	Both Propellers
NOZ	Both Thrust Nozzles
RUDDER	Both Rudders
AERO	Aero Dynamics - Windage
AIR	Air Momentum
GRAV	Gravity
SKIRT	Skirt and Spray Drag
DUCT	Both Propeller Ducts - Windage
CUSH	Cushion Pressure
DAMP	Yaw Aerodynamic Damping
SEA	Seaway and Vehicle Generated Waves

MAIN PROPELLERS. The thrust acting on the vehicle due to each of 2 propellers is calculated and then the total forces and moments for these propellers are calculated.

The thrust of one propeller is plotted by the real-time subroutine at maximum propeller speed and for different head wind velocities versus propeller pitch and shown in Figure 3.

Subroutine: PROPS

PROPELLERS (Forces & Moments)

$$T_{PROPj} = \begin{cases} \left[338\,\beta_{Pj} + 4.36\,\beta^{2}_{Pj} - .1715\,u^{2}_{WIND} - 1.43\,\beta_{Pj}\,u_{WIND}\right]\left[\dfrac{N_{Pj}}{N_{P\,Max}}\right]^{2} & \beta_{Pj} \geq 0 \\ \left[60\,\beta_{Pj} - .1715\,u_{WIND}^{2}\right]\left[\dfrac{N_{Pj}}{N_{P\,Max}}\right]^{2} & \beta_{Pj} < 0 \end{cases}$$

$$X_{PROP} = T_{PROP\,1} + T_{PROP\,2} = T_{PROPS} + T_{PROP\,P}$$

$$M_{PROP} = X_{PROP}\,Z_{P} = 0$$

$$N_{PROP} = -\left[T_{PROP}\,Y_{PS} + T_{PROP\,P}\,Y_{PP}\right]$$

j	1 for STBD, 2 for PORT
T_{PROPj}	Thrust of Propeller j (Lbs)
u_{WIND}	Apparent Head Wind Velocity along X_b (Ft/Sec)
β_{Pj}	Pitch Angle of Propeller j (DEG)
N_{Pj}	Speed of Propeller j (RPM)
Z_{P}	Position of Propeller along Z_b = 0 Ft.
Y_{PS}	Position of STBD Propeller along Y_b = -4.0 Ft.
Y_{PP}	Position of Port Propeller along Y_b = -32.0 Ft.
$N_{P\,Max}$	Maximum Propeller Speed = 1250 RPM

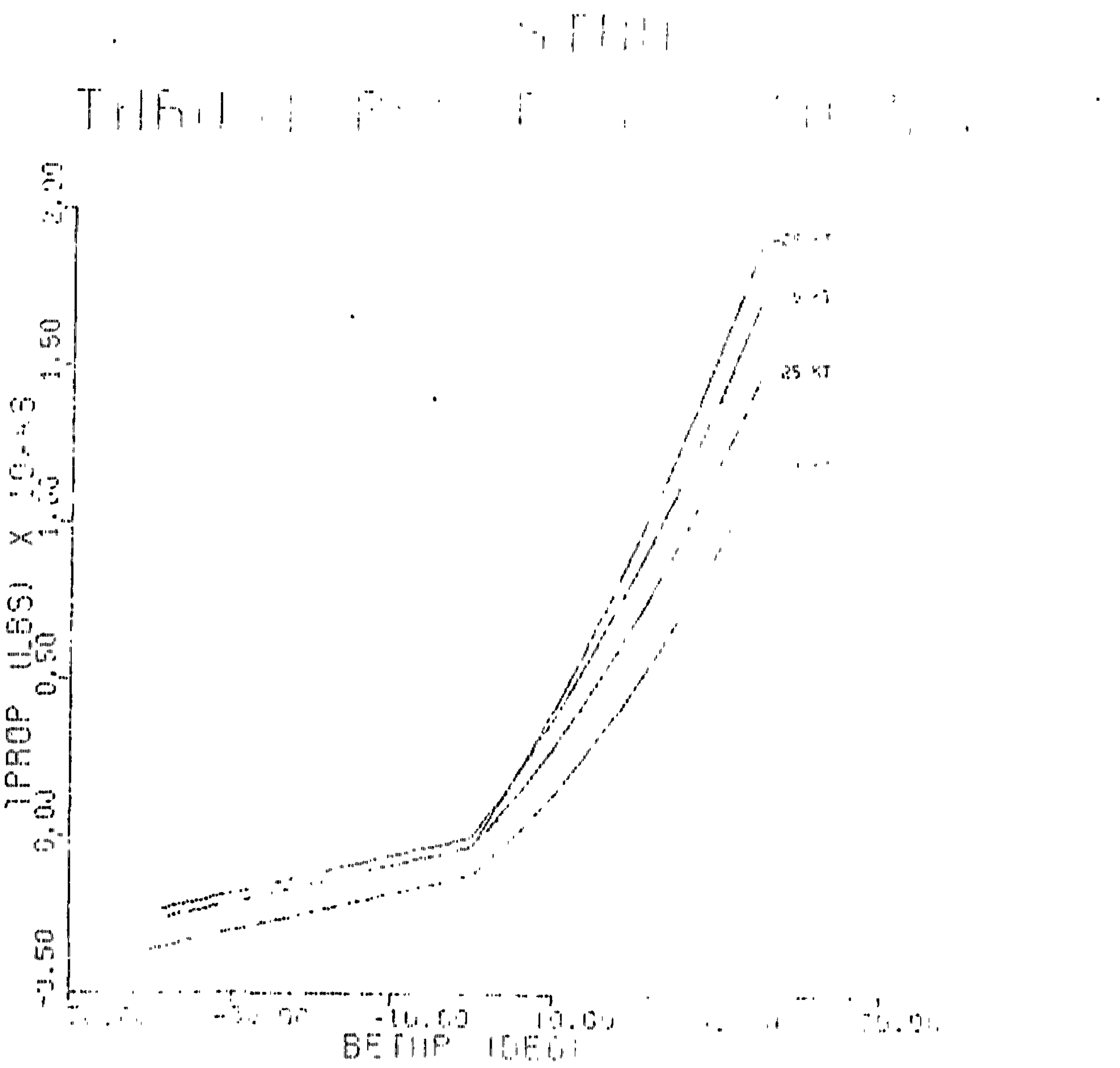


Figure 3. Propeller Thrust

RUDDERS. First, the axial velocity of air through each propeller duct is calculated. Second, the axial air pressure exerted at the rudders is calculated. Third, the total lift and drag coefficients are calculated based on rudder angle. Then, the total forces and moments due to the rudders are calculated.

Subroutine: RUDDERS

RUDDERS (Forces & Moments)

$$V^{2}_{AXIAL\ j} = \begin{cases} (V_{AWIND} \cos\beta_a)^2 + \dfrac{T_{PROP\ j}}{\rho A_{DUCT}} & T_{PROP\ j} \geq 0 \\ (V_{AWIND} \cos\beta_a)^2 + \dfrac{T_{PROPj}}{2\rho A_{DUCT}} & T_{PROP\ j} < 0 \end{cases}$$

$$P_{AXIAL\ j} = 1/2\ \rho V^{2}_{AXIAL\ j}$$

$$C_{LIFT} = \begin{cases} .053\ \psi_R & |\psi_R| \leq 20^{\circ} \\ 1.06\ \dfrac{\psi R}{|\psi_R|} & |\psi_R| > 20^{\circ} \end{cases}$$

$$C_{DRAG} = .02 + (.422 \times 10^{-3})\,\psi_R^{\,2}$$

$$X_{RUDDER} = -C_{DRAG}\ A_{RUDDER}\ (P_{AXIALS} + P_{AXIAL\ P})$$

$$Y_{RUDDER} = C_{LIFT}\ A_{RUDDER}\ (P_{AXIALS} + P_{AXIALP})$$

$$K_{RUDDER} = Z_R\ Y_{RUDDER} \qquad = 0$$

$$M_{RUDDER} = Z_R\ X_{RUDDER} \qquad = 0$$

$$N_{RUDDER} = C_{DRAG}\ A_{RUDDER}\ (Y_{RS}\ P_{AXIALS} + Y_{RP}\ P_{AXIAL\ P}) + X_R Y_{RUDDER}$$

V_{AWIND}	Apparent Wind Velocity (Ft/Sec)
β_a	Apparent Wind Angle (DEG)
T_{PROPj}	Thrust on Propeller j (Lbs)
j	1 for STBD, 2 for PORT
ρ	Air Density = .00237 Slugs/Ft^3
A_{DUCT}	Duct Area = 123 Ft^2

$V_{AXIAL\ j}$	Axial Velocity Inside Duct at Rudder j (Ft/Sec)
$P_{AXIAL\ j}$	Dynamic Pressure Inside Duct at Rudder j (Lbs/In^2)
ψ_R	Rudder Angle (RAD)
C_{LIFT}	Lift Coefficient
C_{DRAG}	Drag Coefficient
A_{RUDDER}	Rudder Area = 47.2 Ft^2
Z_R	Position of Rudder along Z_b = 0 Ft.
Y_{RS}	Position of Rudder along Y_b STBD = -4 Ft
Y_{RP}	Position of Rudder along Y_b PORT = -32 Ft
X_R	Position of Rudder along X_b = -67.1 Ft.

THRUST NOZZLES. The thrust due to each nozzle is calculated and then using the nozzle angle the total forces and moments are calculated due to both nozzles.

Subroutine: NOZZLES

Thrust Nozzles (Forces & Moments)

$$T_{NOZ\ j} = (2.44 \times 10^{-4})\, Q^{2}_{NOZ\ j}$$

$$X_{NOZ} = (T_{NOZS} + T_{NOZ\ P}) \cos \psi_N$$

$$Y_{NOZ} = (T_{NOZS} + T_{NOZ\ P}) \sin \psi_N$$

$$K_{NOZ} = -Y_{NOZ} Z_N$$

$$M_{NOZ} = X_{NOZ} Z_N$$

$$N_{NOZ} = Y_{NOZ} X_N - (T_{NOZS} Y_{NS} + T_{NOZ\ P} Y_{NP}) \cos \psi_N$$

j	1 for STBD, 2 for PORT
$Q_{NOZ\ j}$	Flow Through Nozzle j (Ft^3/Sec)
$T_{NOZ\ j}$	Thrust on Nozzle j (Lbs)
ψ_N	Position of Nozzle (DEG)
X_N	Position of Nozzle along X_b=-23.16 Ft.
Z_N	Position of Nozzle along Z_b=-3.0 Ft.
Y_{NS}	Position of Nozzle along Y_b=-.25 Ft STBD
Y_{NP}	Position of Nozzle along Y_b=-35.75 Ft PORT

PROPELLER DUCTS. First, the wind angle of attack of each duct is calculated. Second, the side force drag coefficient is calculated for each duct. Third, the wind velocity acting on each duct is calculated. Fourth, the wind velocity causes pressure on each duct. Fifth, the total forces and moments on both propeller ducts are calculated.

A Computer plot by the real-time subroutine of the side force coefficient is shown in Figure 4 for verification purposes.

Subroutine: DUCTS

PROPELLER DUCTS- WINDAGE (Forces & Moments)

$$\delta_{DUCT\ j} = \begin{cases} \tan^{-1} \dfrac{V_{AWIND} \sin \beta_a - X_R r}{V_{AWIND} \cos \beta_a + 1.78\sqrt{|T_{PROPj}|}} & T_{PROPj} \geq 0 \\ 0 & T_{PROPj} < 0 \end{cases}$$

$$C_{DUCTj} = \begin{cases} -7.407 \times 10^{-3} \delta_{DUCTj} + 1.333 & \delta_{DUCTj} \geq 18^{\circ} \\ 6.667 \times 10^{-2} \delta_{DUCTj} & \delta_{DUCTj} < 18^{\circ} \end{cases}$$

$$V^2_{DUCTj} = \begin{cases} V^2_{AWIND} + 3.1684\ T_{PROPj} + 3.56 V_{AWIND} \cos \beta_a \sqrt{|T_{PROPj}|} & T_{PROPj} \geq 0 \\ V^2_{AWIND} + .7921\ T_{PROPj} - 1.78 V_{AWIND} \cos \beta_a \sqrt{|T_{PROPj}|} & T_{PROPj} < 0 \end{cases}$$

$$P_{DUCTj} = .5 \rho V^2_{DUCTj}$$

$$Y_{DUCT} = -D_{DUCT}\, \ell_{DUCT} \left(C_{DUCTS} P_{DUCTS} + C_{DUCT\ P} P_{DUCT\ P}\right)$$

$$K_{DUCT} = Y_{DUCT} Z_R = 0$$

$$N_{DUCT} = Y_{DUCT}\, X_R$$

V_{AWIND}	Apparent Wind Velocity (Ft/Sec)
β_a	Apparent Wind Angle (DEG)
r	Angular Velocity about Axis Z_b(RAD/SEC)
T_{PROPj}	Thrust of Propeller j(Lbs)
j	1 for STBD, 2 for PORT
$\delta_{DUCT\ j}$	Angle of Attack - Relative Wind to Duct (DEG)
$V_{DUCT\ j}$	Air Velocity Around Duct (Ft/Sec)
ρ	Air Density = .00237 Slugs/Ft^3
$P_{DUCT\ j}$	Dynamic Pressure Acting on Duct (Lbs/Ft^2)
$C_{DUCT\ j}$	Duct Side Force Coefficient
$D_{DUCT\ j}$	Duct Diameter = 11.25 Ft.
ℓ_{DUCT}	Duct Chord = 4.67 Ft.
X_R	Position of Duct along X_b = -67.1 Ft.
Z_R	Position of Duct along Z_b = 0 Ft.

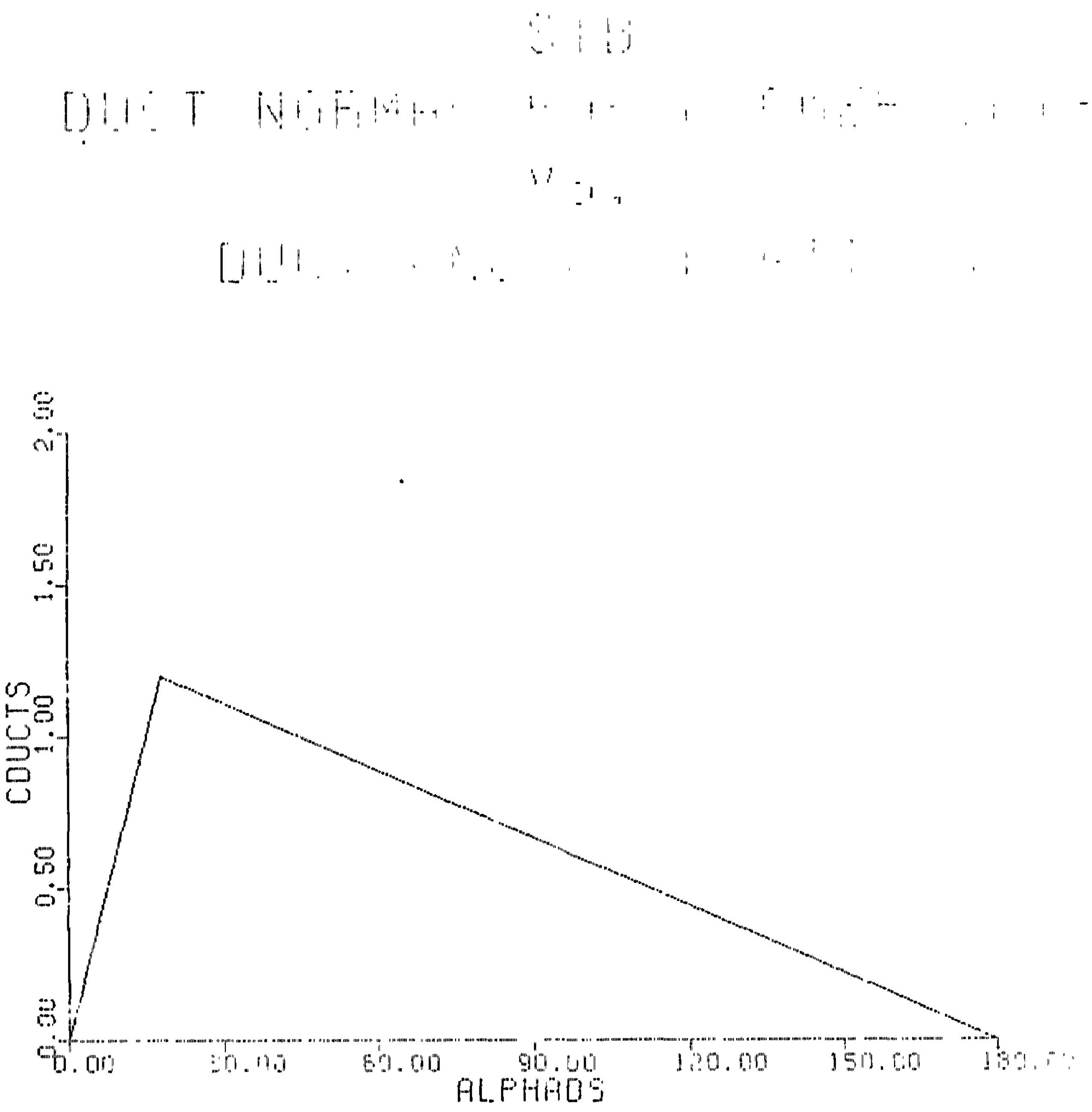


Figure 4.   Propeller Duct Side Force Coefficient

AERODYNAMICS DUE TO WINDAGE. From the apparent wind angle (wind angle of attack acting on the vehicle) the drag coefficients are calculated. The forces and moments are then calculated from the apparent wind velocity.

The aerodynamic drag coefficients are plotted by the real-time subroutine for verification purposes. The sway drag coefficient is shown in Figure 5 . The surge and yaw drag coefficients are shown in Figures 6 and 7 .

Subroutine: AERO

AERODYNAMICS - WINDAGE (Forces & Moments)

$$C_{DRAG\,Y} = \left[.01385\left|\beta_a\right| - 7.69 \times 10^{-5}\left|\beta_a\right|^2\right]\frac{\beta_a}{\left|\beta_a\right|} \qquad .55 \le C_{DRAG\,Y} \le .55$$

$$C_{DRAG\,X} = \begin{cases} -2.22 \times 10^{-4}\left|\beta_a\right|^2 + 3.33 \times 10^{-3}\left|\beta_a\right| + .5 & \left|\beta_a\right| < 40^\circ \\ -2.48 \times 10^{-5}\left|\beta_a\right|^3 + 5 \times 10^{-3}\left|\beta_a\right|^2 - .324\left|\beta_a\right| + 6.835 & 40 \le \left|\beta_a\right| < 88^\circ \\ 1.04 \times 10^{-4}\left|\beta_a\right|^2 - 3.518 \times 10^{-2}\left|\beta_a\right| + 2.39 & \left|\beta_a\right| \ge 88^\circ \end{cases}$$

$$C_{DRAG\,N} = \begin{cases} \left[1.67 \times 10^{-3}\left|\beta_a\right|\right]\frac{\beta_a}{\left|\beta_a\right|} & \left|\beta_a\right| < 60^\circ \\ \left[5.17 \times 10^{-7}\left|\beta_a\right|^3 - 1.23 \times 10^{-4}\left|\beta_a\right|^2 + 5.82 \times 10^{-3}\left|\beta_a\right| + 8.27 \times 10^{-2}\right]\frac{\beta_a}{\left|\beta_a\right|} & 60^\circ \le \left|\beta_a\right| \le 120^\circ \\ \left[1.67 \times 10^{-3}\left|\beta_a\right| - .3006\right]\frac{\beta_a}{\left|\beta_a\right|} & \left|\beta_a\right| > 120^\circ \end{cases}$$

$$X_{AERO} = -C_{DRAG\,X}A_{FRONT} \cdot .5\rho V^2_{AWIND}$$

$$Y_{AERO} = -C_{DRAG\,Y}A_{PLAN} \cdot .5\rho V^2_{AWIND}$$

$$K_{AERO} = -Y_{AERO}Z_{CG}$$

$$M_{AERO} = X_{AERO}Y_{CG}$$

$$N_{AERO} = C_{DRAG\,N}A_{PLAN}\ell_{CUSH} \cdot .5\rho V^2_{AWIND} + Y_{AERO}X_{CG} - X_{AERO}Y_{CG}$$

β_a	Apparent Wind Angle (DEG)
$C_{DRAG\ Y}$	Side Force Drag Coefficient
$C_{DRAG\ X}$	Frontal Drag Coefficient
$C_{DRAG\ N}$	Yaw Moment Coefficient
A_{FRONT}	Frontal Area = 836 Ft^2
ρ	Air Density = .00237 Slugs/Ft^3
V_{AWIND}	Apparent Wind Velocity (Ft/Sec)
X_{CG}	Position of CG measured along X_b = -30.0 Ft.
Y_{CG}	Position of CG measured along Y_b = -18.0 Ft.
Z_{CG}	Position of CG measured along Z_b = + 8.0 Ft.
A_{PLAN}	Planform Area = 3200 Ft^2
ℓ_{CUSH}	Length of Cushion = 77 ft.

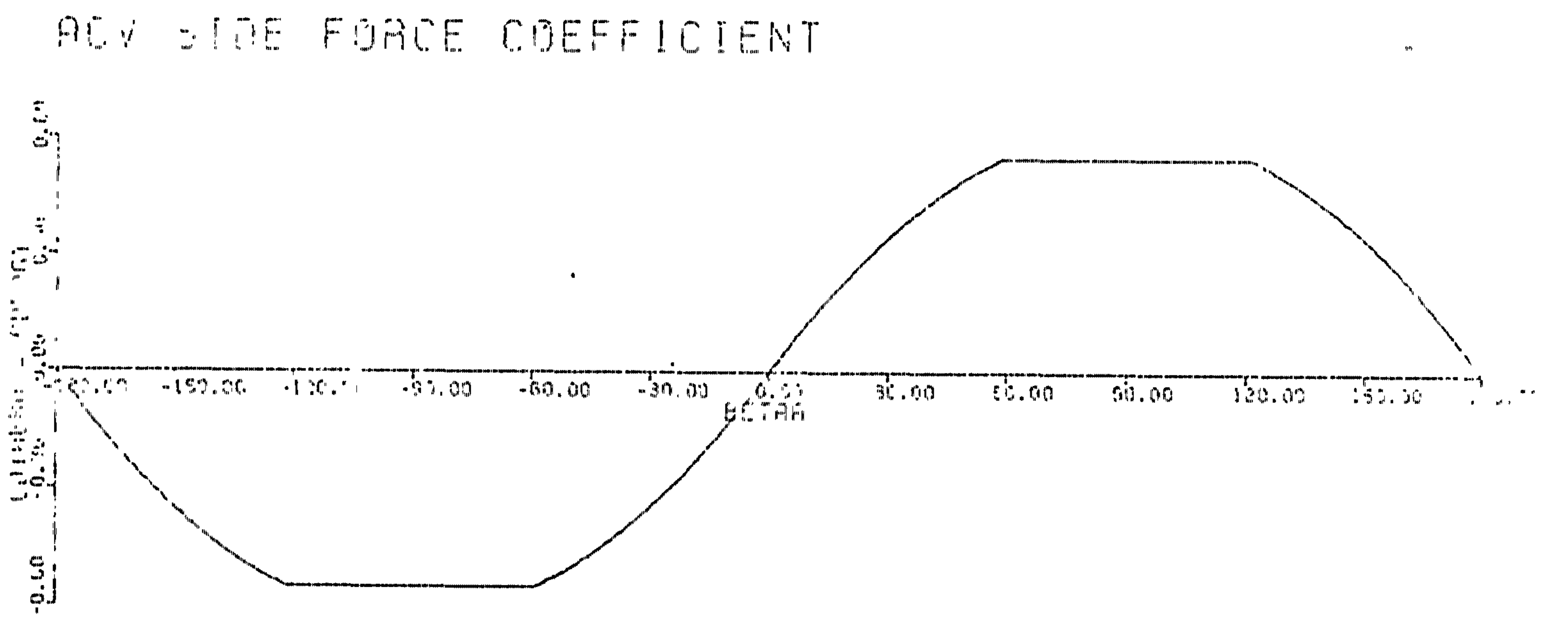


Figure 5. Aerodynamic Drag-Sway Coefficient

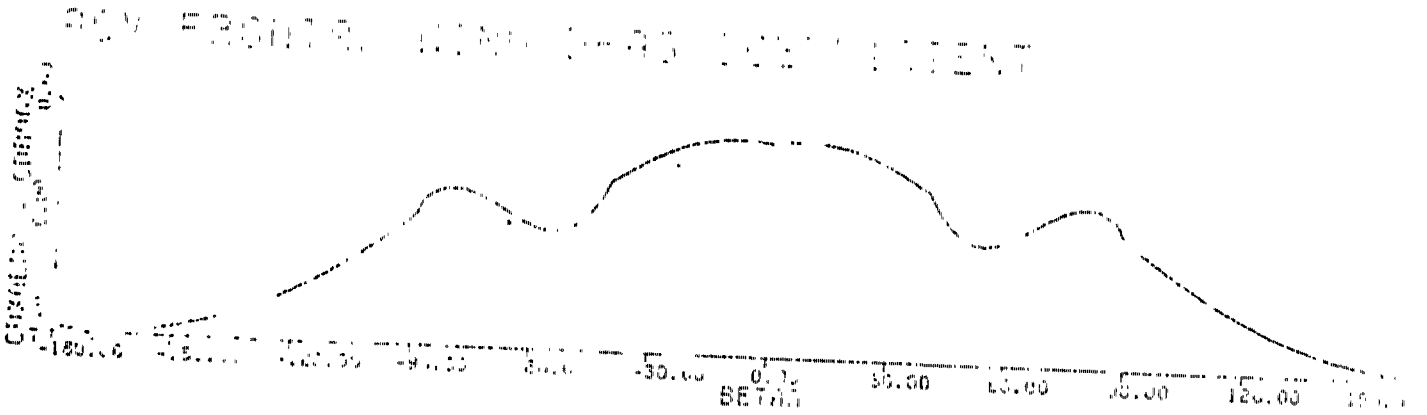


Figure 6. Aerodynamic Drag-Surge Coefficient

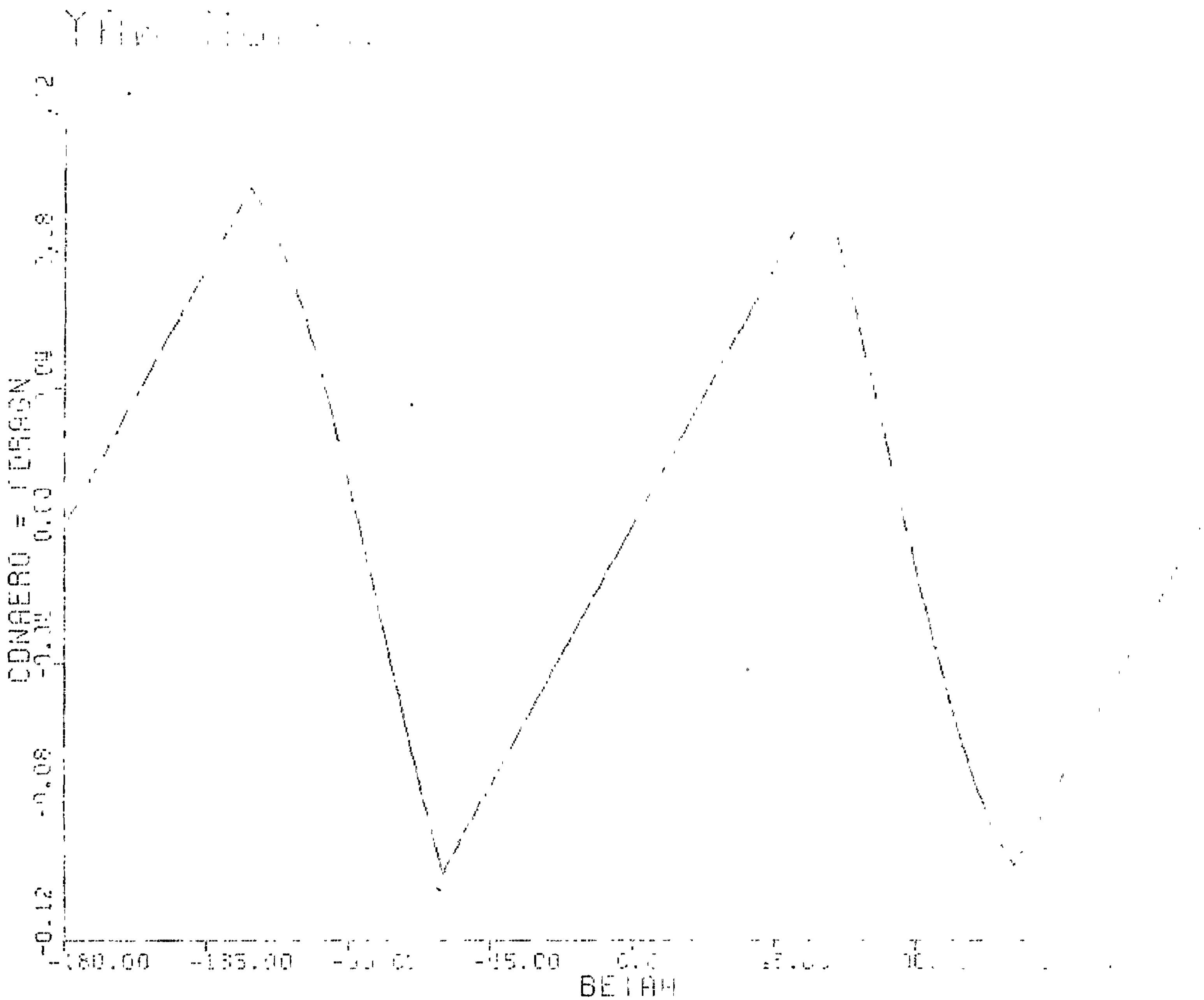


Figure 7 . Aerodynamic Drag-Yaw Coefficient

YAW RATE DAMPING MOMENT. Using a constant drag coefficient, the damping moment is calculated from the yaw rate.

Subroutine: DAMP

Yaw Rate Damping (Forces & Moments)

$N_{DAMP} = -2.77 \times 10^6 r$

r Angular Velocity about Axis Z_b (Rad/Sec)

GRAVITY. From the roll and pitch angles and weight of the vehicle the total forces and moments due to gravity are calculated.

Subroutine: GRAVITY

Gravity (Forces & Moments) (When Integrating, include with Equations of Motion)

$$X_{GRAV} = -Z_{GRAV}\theta$$

$$Y_{GRAV} = Z_{GRAV}\phi$$

$$Z_{GRAV} = mg$$

$$K_{GRAV} = Z_{GRAV}Y_{CG} - Y_{GRAV}Z_{CG}$$

$$M_{GRAV} = X_{GRAV}Z_{CG} - Z_{GRAV}X_{CG}$$

$$N_{GRAV} = Y_{GRAV}X_{CG} - X_{GRAV}Y_{CG}$$

m	ACV Mass = 10879.5 Slugs
g	Acceleration of Gravity = 32.17 Ft/Sec ^{2}
X_{CG}	Position of CG measured along X_b = -30.0 Ft
Y_{CG}	Position of CG measured along Y_b = -18.0 Ft
Z_{CG}	Position of CG measured along Z_b = +8.0 Ft
ϕ	Euler Angle Roll of ACV (Rad)
θ	Euler Angle Pitch of ACV (Rad)

AIR MOMENTUM. The air momentum force is due to an unbalanced flow through the starboard and port fans caused by an apparent wind velocity. The total forces and moments are then calculated.

Subroutine: AIR
Air Momentum (Forces and Moments)

$$F_{AIR} = -\rho V_{AWIND} \left(Q_{FANS} + Q_{FAN\ P} \right)$$

$$X_{AIR} = F_{AIR} \cos \beta_{a}$$

$$Y_{AIR} = F_{AIR} \sin \beta_{a}$$

$$K_{AIR} = -Z_{CG} Y_{AIR}$$

$$M_{AIR} = Z_{CG} X_{AIR}$$

$$N_{AIR} = (X_{CG} - X_{F}) Y_{AIR} - Y_{CG} X_{AIR}$$

ρ	Air Density = .00237 Slugs/Ft ^{3}
V_{AWIND}	Apparent Wind Velocity (Ft/Sec)
β_{a}	Apparent Wind Angle (DEG)
Q_{FANS}	Total Flow Through STBD Fan (Ft^3/Sec)
$Q_{FAN\ P}$	Total Flow Through PORT Fan (Ft^3/Sec)
F_{AIR}	Force Acting on ACV due to Air Drag (Lbs)
X_{CG}	Position of CG measured along X_b = -30.0 Ft
Y_{CG}	Position of CG measured along Y_b = -18.0 Ft
Z_{CG}	Position of CG measured along Z_b = +8.0 Ft
X_{F}	Position of Fans measured along X_b = -12.0 Ft

CUSHION PRESSURE. Air pressure under the hull causes the primary lift of the vehicle. Unbalances in pressures cause moments to be applied to the vehicle.

Subroutine: CUSH

CUSHION PRESSURE (Forces & Moments)

$$Z_{CUSH} = -A_{CUSH} \sum_{i=1}^{4} P_{CUSH\ i}$$

$$K_{CUSH} = -A_{CUSH} \sum_{i=1}^{4} P_{CUSH\ i} Y_{ci}$$

$$M_{CUSH} = A_{CUSH} \sum_{i=1}^{4} P_{CUSH\ i} X_{ci}$$

A_{CUSH} Cushion Compartment Area = 800 Ft^2

i=1 STBD Fwd Cushion Compartment

i=2 STBD Aft Cushion Compartment

i=3 PORT Aft Cushion Compartment

i=4 PORT Fwd Cushion Compartment

X_{ci} Position of Center of Cushion Compartment along X_b

Y_{ci} Position of Center of Cushion Compartment along Y_b

$P_{CUSH\ i}$ Pressure within Cushion Compartment (Lbs/Ft^2)

X_{c1} = -10 Ft.

X_{c2} = -50 Ft.

X_{c3} = -50 Ft.

X_{c4} = -10 Ft.

Y_{c1} = -8 Ft.

Y_{c2} = -8 Ft.

Y_{c3} = -28 Ft.

Y_{c4} = -28 Ft.

VEHICLE GENERATED WAVE DRAG APPROXIMATION. A constant drag coefficient of .5 in both longitudinal and lateral directions on the vehicle is used to calculate v^2 drag forces and moments. This approximation is only used when the vehicle generated wave package is not running. Primarily, this approximation was made to facilitate development of the simulation.

Subroutine: SKIRT

WAVE - CUSHION PRESSURE DRAG APPROXIMATION (Forces & Moments) (For use without Vehicle Generated Waves)

$$X_{SKIRT} = -.5\,u\,|u|$$

$$Y_{SKIRT} = -.5\,v\,|v|$$

$$K_{SKIRT} = -Z_{CG} Y_{SKIRT}$$

$$M_{SKIRT} = -Z_{CG} X_{SKIRT}$$

$$N_{SKIRT} = X_{CG} Y_{SKIRT} - Y_{CG} X_{SKIRT}$$

u Velocity (Linear) in direction X_b (Ft/Sec)

v Velocity (Linear) in direction Y_b (Ft/Sec)

X_{CG} Position of CG measured along $X_b = -30.0$ Ft

Y_{CG} Position of CG measured along $Y_b = -18.0$ Ft

Z_{CG} Position of CG measured along $Z_b = +8.0$ Ft

SKIRT AND SPRAY DRAG. The composite effects of the skirt taking wave slap and spray hitting the bow and sides of the vehicle are taken into consideration by a v^2 type of drag with constant coefficient of .25. Over land, there is no wave slap and spray, so the drag is zero. The forces and moments are then calculated.

Subroutine: SKIRT VG

SKIRT AND SPRAY DRAG (Forces & Moments)
(For use with Vehicle Generated Waves)

$$H_P = h_{w4} + Z_{HULL}$$

$$X_{SKIRT} = -C_D \quad u \left| u \right|$$

$$Y_{SKIRT} = -C_D \quad v \left| v \right|$$

$$K_{SKIRT} = -Y_{SKIRT} \, H_P$$

$$M_{SKIRT} = X_{SKIRT} \, H_P$$

$$N_{SKIRT} = Y_{SKIRT} \, X_{CG} - X_{SKIRT} \, Y_{CG}$$

C_D	Drag Coefficient = .25, = 0 over land
u	Vehicle Surge Velocity along X_b (Ft/Sec)
v	Vehicle Sway Velocity along Y_b (Ft/Sec)
H_P	Height of Pilot Cabin above water (Ft)
X_{CG}	Position of CG measured along X_b = -30.0 Ft.
Y_{CG}	Position of CG measured along Y_b = -18.0 Ft.
h_{w4}	Height over water of point No. 4 (Ft)
Z_{HULL}	Height of Pilot Cabin above Hull Bottom = 12 Ft.

SEAWAY AND VEHICLE GENERATED WAVES. The sum of the wave heights at specific points under the hull of the waves before the vehicle arrived (seaway) and the waves caused by the vehicle (VGW) is used by a differencing method to arrive at the total forces and moments pushing on the skirt of the vehicle.

Subroutine: SEA

Seaway and Vehicle Generated Waves (Forces & Moments)

$$X_{SEA} = -\frac{20}{3}\Big[P_{CUSH\ 1}(\eta_1+\eta_2+\eta_3-\eta_5-\eta_{20}-\eta_{18})$$

$$+P_{CUSH\ 2}(\eta_5+\eta_{20}+\eta_{18}-\eta_7-\eta_8-\eta_9)$$

$$+P_{CUSH\ 3}(\eta_{18}+\eta_{21}+\eta_{13}-\eta_9-\eta_{10}-\eta_{11})$$

$$+P_{CUSH\ 4}(\eta_1+\eta_{16}+\eta_{15}-\eta_{18}-\eta_{21}-\eta_{13})\Big]$$

$$Y_{SEA} = -\frac{40}{3}\Big[P_{CUSH\ 1}(\eta_3+\eta_4+\eta_5-\eta_1-\eta_{17}-\eta_{18})$$

$$+P_{CUSH\ 2}(\eta_5+\eta_6+\eta_7-\eta_9-\eta_{19}-\eta_{18})$$

$$+P_{CUSH\ 3}(\eta_9+\eta_{19}+\eta_{18}-\eta_{11}-\eta_{12}-\eta_{13})$$

$$+P_{CUSH\ 4}(\eta_1+\eta_{17}+\eta_{18}-\eta_{13}-\eta_{14}-\eta_{15})\Big]$$

$$N_{SEA} = P_{CUSH\ 1}\Big[\frac{(20)(8)}{3}(\eta_5+\eta_{20}+\eta_{18}-\eta_1-\eta_2-\eta_3)$$

$$+\frac{(40)(10)}{3}(\eta_3+\eta_4+\eta_5-\eta_1-\eta_{17}-\eta_{18})\Big]$$

$$+P_{CUSH\ 2}\Big[\frac{(40)(50)}{3}(\eta_5+\eta_6+\eta_7-\eta_9-\eta_{19}-\eta_{18})$$

$$+\frac{(20)(8)}{3}(\eta_7+\eta_8+\eta_9-\eta_5-\eta_{20}-\eta_{18})\Big]$$

$$+P_{CUSH\ 3}\Big[\frac{(20)(28)}{3}(\eta_9+\eta_{10}+\eta_{11}-\eta_{18}-\eta_{21}-\eta_{13})$$

$$+\frac{(40)(50)}{3}(\eta_9+\eta_{19}+\eta_{18}-\eta_{11}-\eta_{12}-\eta_{13})\Big]$$

$$+P_{CUSH\ 4}\Big[\frac{(20)(28)}{3}(\eta_{18}+\eta_{21}+\eta_{13}-\eta_1-\eta_{16}-\eta_{15}$$

$$+\frac{(40)(10)}{3}(\eta_1+\eta_{17}+\eta_{18}-\eta_{13}-\eta_{14}-\eta_{15}\Big]$$

i	1-21, Cushion Periphery Points
η_i	Total height of waves due to seaway and VGW (Ft)
$P_{CUSH\ 1}$	Pressure within compartment 1 (Lbs/Ft^2)
$P_{CUSH\ 2}$	Pressure within compartment 2 (Lbs/Ft^2)
$P_{CUSH\ 3}$	Pressure within compartment 3 (Lbs/Ft^2)
$P_{CUSH\ 4}$	Pressure within compartment 4 (Lbs/Ft^2)

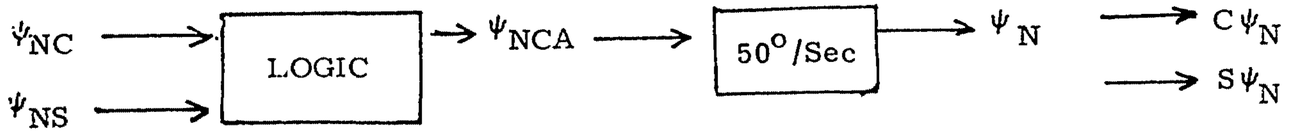
PILOT CONTROLS

The pilot controls are the devices that the pilot manipulates in order to properly maneuver the vehicle.

THRUST NOZZLES. The pilot has a steering wheel that directs the thrust nozzle positions ±90°. He, also, has a switch to direct the thrust forward or aft. These two pilot commands are used to generate an actual nozzle angle command that sweeps 360°. Using a constant rate servo system, the nozzles are rotated to obtain the actual nozzle angle. (See Figure 8).

Subroutine: EFPOS

NOZZLE CONTROL



$\psi_{NCA} = \psi_{NC}$ $\qquad \psi_{NS} = '1'$

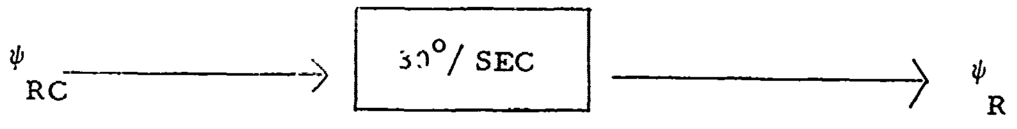
$$\psi_{NCA} = \begin{cases} 180 - \psi_{NC} & \psi_{NS} = '0',\ \psi_{NC} \geq 0 \\ -180 - \psi_{NC} & \psi_{NC} < 0 \end{cases}$$

ψ_{NC}	Nozzle angle command ($\pm$ 90°)
ψ_{NS}	Nozzle angle switch ('1' for -90° to 90°) or ('0' for 90° to 270°)
ψ_N	Nozzle angle ($\pm$ 180°)
$C\psi_N$	Cos ψ_N
$S\psi_N$	Sin ψ_N

Figure 8 . Thrust Nozzle Control

RUDDERS. Foot pedals activated by the pilot command a rudder angle. Through a constant rate servo system the rudder angle is obtained. (See Figure 9 .)

RUDDER CONTROL



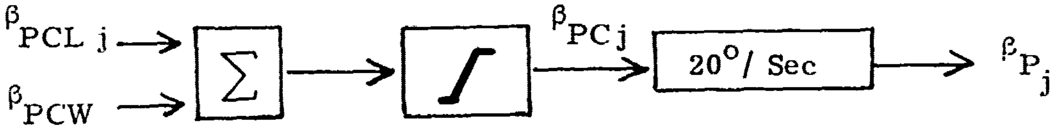
ψ_{RC} Rudder Angle Command ± 30°

ψ_R Rudder Angle ± 30°

Figure 9 . Rudder Control

PROPELLER PITCH. The pilot has the controls to be able to command a change in pitch of each propeller. The pilot, also, has the ability to change the pitch up to 20° on both propellers by a vernier located on the steering wheel control stick. Through a constant rate servo on each propeller, the pitch angle of each propeller is obtained. (See Figure 10.)

PROPELLER PITCH CONTROL



j	1 for STBD, 2 for PORT
β_{PCW}	Vernier Wheel Command (+ 10° to -10°)
β_{PCLj}	Propeller Pitch Angle Command from pilot (± 40°)
β_{PCj}	Propeller Pitch Angle Command (+35° to -40°)
β_{Pj}	Propeller Pitch Angle (+35° to -40°)

Figure 10 . Propeller Pitch Control

TURBINES. The pilot has the controls to vary all 6 gas turbine speeds and to independently control the power shaft speeds. The speeds of the gas turbines are determined by a first order lag with a time constant of 3 seconds. However, the speed of the power shaft is governor controlled and if set too low will tend to decrease the operating speeds of the gas turbines through a first order lag of 2 seconds.

For the particular operating speeds of the gas turbines and power shafts, the optimum speed of the power shaft and the gas turbine optimum shaft horsepower are calculated. From these the actual shaft horsepower of each gas turbine is computed and it is a simple calculation to arrive at the shaft horsepower for each power shaft. The power absorbed by each propeller and fan is determined from its characteristics. The angular acceleration of each power shaft is then calculated and integrated to obtain the power shaft speed, port and starboard. The fan and propeller speeds are geared directly to the port and starboard power shafts.

Subroutine: TURBINES

TURBINES

$$N_{Tn} = \left\{ \begin{array}{l} N_{Tn} + S_T \, (N_{TCn} - N_{Tn}) \ \Delta t \\ N_{Tn} + S_S \, (N_{SCj} - N_{Sj}) \Delta t \end{array} \right\} \geq 12125 \qquad \begin{array}{l} N_{Sj} < N_{SCj} \\ N_{Sj} \geq N_{SCj} \end{array}$$

$$N_{SOPTn} = 11672 - 1.90457 \, N_{Tn} + 1.21488 \times 10^{-4} \, N_{Tn}{}^{2}$$

$$SHP_{OPTn} = 14684 - 2.3976 \, N_{Tn} + 9.796 \times 10^{-5} \, N_{Tn}{}^{2}$$

$$SHP_n = SHP_{OPTn} \left[.1 + 1.8 \left(\frac{N_{sj}}{N_{SOPTn}} \right) - .9 \left(\frac{N_{sj}}{N_{SOPTn}} \right)^{2} \right] \leq \frac{3500}{12200} N_{Sj}$$

$$SHP_{SS} = SHP_{S1} + SHP_{S2} + SHP_{S3}$$

$$SHP_{SP} = SHP_{P1} + SHP_{P2} + SHP_{P3}$$

$$C_{Pj} = \left\{ \begin{array}{ll} 0 & \\ \frac{52}{127.5} \quad u_{WIND} \, (10 - \beta_{Pj}) & \end{array} \right. \qquad \begin{array}{l} \beta_{Pj} \geq 10 \\ \beta_{Pj} < 10 \end{array}$$

$$HP_{PROPj} = \left(\frac{N_{Pj}}{1250} \right)^{3} \quad (450 + 23.05 \beta_{Pj} + 2.56 \beta_{Pj}{}^{2} + C_{Pj})$$

$$HP_{FAN\,j} = \frac{P_{MAN\,j} \; Q_{FAN\,j}}{540 \quad \eta_{FAN}}$$

$$\dot{N}_{Sj} = \frac{SHP_{Sj} - HP_{FAN\,j} - HP_{PROP\,j}}{N_{Sj} \; I_{MACH}} \quad \frac{33000}{2\pi} \quad \frac{60}{2\pi}$$

$$N_{Sj} = N_{Sj} + \dot{N}_{Sj} \, \Delta \, t \geq 1$$

$$N_{FAN\,j} = .1297 \, N_{Sj}$$

$$N_{Pj} = .6427 \, N_{FAN\,j}$$

S_T	Time Constant of Turbine Throttle = .3 Sec $^{-1}$
S_S	Time Constant of Power Shaft Control = .5 Sec $^{-1}$
Δt	Sampling Period (Sec)
j	1, n = 1, 2, 3 STARBOARD
	2, n = 4, 5, 6 PORT
N_{Tn}	Turbine Speed (RPM)
N_{TCn}	Turbine Speed Command (RPM) $(15500 \leq N_{TCn} \leq 18700)$
N_{Sj}	Power Shaft Speed (RPM)
N_{SCj}	Power Shaft Speed Command (RPM) $(9000 \leq N_{SCj} \leq 16000)$
N_{SOPTn}	Optimum Power Shaft Speed for each Turbine (RPM)
SHP_{OPTn}	Optimum Shaft Horsepower for each Turbine (HP)
SHP_n	Actual Horsepower Output for each Turbine (HP)
TQ_{Tn}	Torque produced for each Turbine (Ft-Lbs)
TQ_{SS}	Power Shaft Torque STBD (Ft-Lbs)
TQ_{SP}	Power Shaft Torque PORT (Ft-Lbs)
$HP_{FAN\ j}$	Power Absorbed by Cushion Fans (HP)
$P_{MAN\ j}$	Manifold Pressures (Lbs/Ft^2)
$Q_{FAN\ j}$	Fan Air Flows (Ft^3/Sec)

η_{FAN}	Fan Efficiency = .85
β_{Pj}	Pitch Angle of Propeller (DEG)
u_{WIND}	Apparent Head Wind Velocity along X_b (Ft/Sec)
C_{Pj}	Calculated Constant
$HP_{PROP\ j}$	Power Absorbed by Propellers (HP)
$TQ_{FAN\ j}$	Torque used by Fans (Ft-Lbs)
$TQ_{PROP\ j}$	Torque used by Propellers (Ft-Lbs)
$\dot{N}_{Sj}$	Power Shaft Accelerations (RPM/Sec)
I_{MACH}	Machinery Moment of Inertia = 3.533 Slug-Ft ^{2}
$N_{FAN\ j}$	Fan Speeds (RPM)
N_{Pj}	Propeller Speeds (RPM)

COCKPIT CONTROLS AND DISPLAYS - TEMPORARY. For the temporary cockpit mockup at NAVTRAEQUIPCEN for which this simulation was developed, all controls were implemented. However, the gas turbine controls were lumped together and divided PORT, STBD. On the vehicle there are 3 gas turbines on each port and starboard side. Instead of 6 individual throttles only 2 were provided so that the 3 turbines on each side receive the same command. Table 3 shows the implemented controls and how they were scaled.

Not all displays were implemented as they would be on the vehicle. Most meters were synchro driven instead of straight analog voltage type meters. Only 1 of the 6 gas turbine speeds was displayed. Table 4 shows the implemented displays and how they were scaled. Figure 11 shows the layout configuration of the controls and displays which were implemented.

TABLF 3 . SCALING FROM COCKPIT CONTROLS - TEMPORARY

INPUT COMMAND	MAX/MIN VOLTAGE	SCALE FACTOR/ 10 VOLTS	MAX/MIN VALUE
Bow Thruster Angle (Steering Wheel)	+ 2.5 - 2.5	+ 496.8	+ 90° - 90°
Vernier Propeller Pitch (Wheel-Stick Fwd-Aft)	+ 9.0 - 9.0	-12.1	- 10° + 10°
Rudder (Pedals)	+ 7.0 - 7.0	+51.42852	+ 30° - 30°
STBD Propeller Pitch	+ 9.0 - 9.0	+44.4444	+ 40° - 40°
PORT Propeller Pitch	+ 9.0 - 9.0	+44.4444	+ 40° - 40°
STBD Power Shaft	+ 9.0 0.0	+ 17776	+ 15,000 RPM + 9,000 RPM
PORT Power Shaft	+ 9.0 0.0	+ 17776	+ 15,000 RPM + 9,000 RPM
STBD Gas Turbines (3)	+ 9.0 0.0	+ 18900	+ 17,200 RPM + 15,500 RPM
PORT Gas Turbines (3)	+ 9.0 0.0	+ 18900	+ 17,200 RPM + 15,500 RPM

Subroutine: DTA

TABLE 4 . SCALING TO COCKPIT DISPLAYS - TEMPORARY

OUTPUT PARAMETER	SCALING	VOLTAGE	MAX/MIN VALUE
STBD Gas Turbine Speed	260° = 100%	+ 7.22 0	+ 18,700 RPM 0 RPM
CRAFT Forward Speed	360° = 100%	± 10	+ 80 KTS - 80 KTS
Heading	360° = 100%	± 10	+ 180° - 180°
Roll	360° = 100%	± 10	+ 20° - 20°
Pitch	360° = 100%	± 10	+ 20° - 20°
Hull Height at CG	324° = 100%	+ 9 0	+ 9 Ft. 0 Ft.
PORT Power Shaft Speed	300° = 100%	+8.33 0	+ 16,000 RPM 0 RPM
STBD Power Shaft Speed	300° = 100%	+ 8.33 0	+ 16,000 RPM 0 RPM
STBD Propeller Pitch		± 10	+ 50° - 50°
PORT Propeller Pitch		± 10	+ 50° - 50°

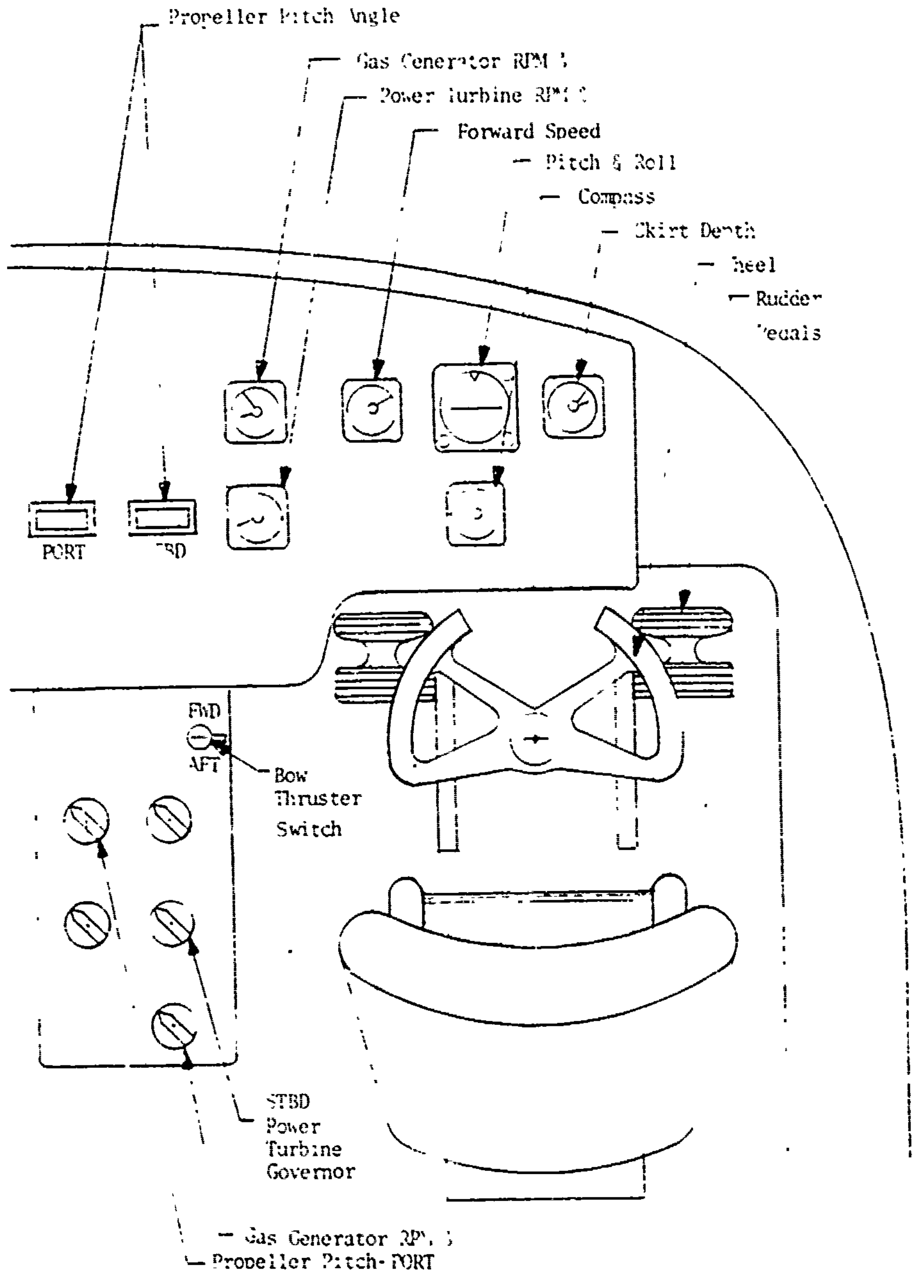


Figure 10. Cockpit Controls and Displays - Temporary

COCKPIT CONTROLS AND DISPLAYS. Since the real-time ACV-LC simulation was developed, NAVTRAEQUIPCEN has put into operation a cockpit which more closely resembles the pilot cabin aboard the ACV-LC. Table 5 shows the implemented controls and how they have been scaled. Table 6 shows the implemented displays and how they were scaled. Figure 12 shows the current configuration of the controls and displays as implemented.

TABLE 5 . SCALING FROM COCKPIT CONTROLS

INPUT COMMAND	MAX/MIN VOLTAGE	SCALE FACTOR/ 10 VOLTS	MAX/MIN VALUE
Bow Thruster Angle (Steering Wheel)	+ 2.1 - 2.1	+ 428	+ 90° - 90°
Vernier Propeller Pitch (Wheel-Stick Fwd-Aft)	+ 2.1 - 2.1	- 47.8	- 10° + 10°
Rudder (Pedals)	+ 2.1 - 2.1	+ 143	+ 30° - 30°
STBD Propeller Pitch	+ 2.5 - 2.9	+ 140	+ 35° - 40°
PORT Propeller Pitch	+ 2.5 - 2.9	+ 140	+ 35° - 40°
STBD Power Shaft	+ 7.4 0.0	+ 20,250	+ 15,000 RPM + 9,000 RPM
PORT Power Shaft	+ 7.4 0.0	+ 20,250	+ 15,000 RPM + 9,000 RPM
STBD Gas Turbines (3)	+ 6.2 0.0	+ 27,692	+ 17,200 RPM + 15,500 RPM
PORT Gas Turbines (3)	+ 6.2 0.0	+ 27,692	+ 17,200 RPM + 15,500 RPM

Subroutine: DTA

TABLE 6 . SCALING TO COCKPIT DISPLAYS

OUTPUT PARAMETER	SCALING	VOLTAGE	MAX/MIN VALUE
PORT Gas Turbine Speed		+ 10 0	+ 17,200 RPM 0 RPM
CRAFT Forward Speed		± 10	+ 80 KTS - 80 KTS
Heading	360° = 100%	± 10	+ 180° - 180°
Pitch		∓ 10	+ 10° - 10°
Hull Height at CG		+ 9 0	+ 9 Ft. 0 Ft.
PORT Power Shaft Speed		+ 10 0	+ 15,000 RPM 0 RPM
STBD Power Shaft Speed		+ 10 0	+ 15,000 RPM 0 RPM
STBD Propeller Pitch		∓ 10	+ 50° - 50°
PORT Propeller Pitch		∓ 10	+ 50° - 50°
North Velocity		± 10	+ 100 Ft/Sec - 100 Ft/Sec
East Velocity		± 10	+ 100 Ft/Sec - 100 Ft/Sec

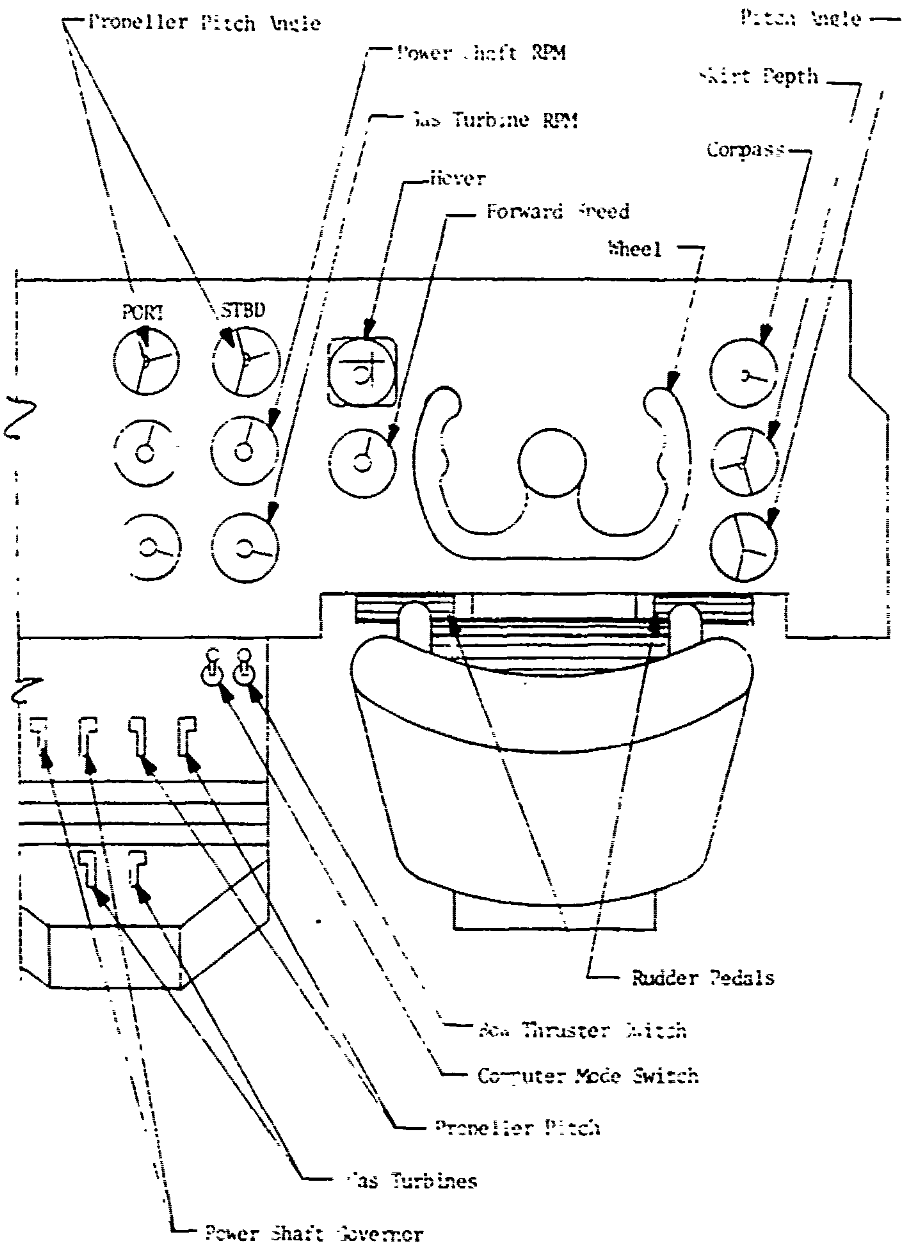


Figure 12. Cockpit Controls and Displays

AIR FLOW AND PRESSURES

Because this vehicle is air cushioned, air is the dependent medium. The flow of air must be directed into the various compartments with sufficient volume and pressure to lift the vehicle above the water surface. The intent of this section is to describe the solution of air pressures in the compartments. This solution is not an easy task because air flow causes air pressure and air pressure causes air flow. Although the intent is to describe the pressures, it is necessary to describe in several sections component parts of the air flow-pressure relationship.

FAN AND THRUST NOZZLE AIR FLOW. Air flow out through each thrust nozzle is calculated directly fror the manifold pressures. Air flow through the fans is calculated from the manifold pressure, but is also dependent upon the fan speed.

Subroutine: AIRFLOW

AIR FLOW RATES

$$Q_{NOZS} = -346 \sqrt{|P_{MANS}|} \; \frac{P_{MANS}}{|P_{MANS}|}$$

$$Q_{NOZ\ P} = -346 \sqrt{|P_{MAN\ P}|} \; \frac{P_{MAN\ P}}{|P_{MAN\ P}|}$$

$$Q_{FANS} = \frac{N_{FANS}}{2000} \left[-640 \sqrt{|P_{MANS}-300|} \; \frac{P_{MANS}-300}{|P_{MANS}-300|} \; -15.8(P_{MANS}-300) \right]$$

$$Q_{FAN\ P} = \frac{N_{FAN\ P}}{2000} \left[-640 \sqrt{|P_{MANP}-300|} \; \frac{P_{MANP}-300}{|P_{MANP}-300|} \; -15.8\ (P_{MANP}-300) \right]$$

P_{MANS}	Pressure in STBD Manifold (Lbs/Ft^2)
P_{MANP}	Pressure in PORT Manifold (Lbs/Ft^2)
N_{FANS}	Angular speed of STBD Fan (RPM)
N_{FANP}	Angular Speed of PORT Fan (RPM)
Q_{NOZS}	STBD Nozzle Flow Rate (Ft^3/Sec)
Q_{NOZP}	PORT Nozzle Flow Rate (Ft^3/Sec)
Q_{FANS}	STBD Fan Flow Rate (Ft^3/Sec)
Q_{FANP}	PORT Fan Flow Rate (Ft^3/Sec)

HEIGHT OF HULL BOTTOM PLATING. The height of each hull bottom plating point above water is calculated from the height above mean water of the pilot, the total wave height (seaway + VGW), the roll and pitch angles, and the X,Y,Z positions of each hull point. Figure 13 shows the definition of the hull bottom plating points as they are located on the planform. Table 7 defines the coordinates of the hull bottom plating points relative to the pilot cabin.

Subroutine: HULL HT

HEIGHT OF HULL BOTTOM PLATING ABOVE WATER

$$H_{HULL\ k} = -Z_i + \theta X_{HULL\ k} - \phi Y_{HULL\ k} - Z_{HULL\ k}$$

$$\eta_k = \eta_{SEA\ k} + \eta_{VGW\ k}$$

$$h_{wk} = H_{HULL\ k} - \eta_k$$

k	Number of Hull Bottom Plating Points
Z_i	Inertial Position (Vertical) of Pilot Cabin (Ft)
ϕ	Euler Angle Roll (RAD)
θ	Euler Angle Pitch (RAD)
$X_{HULL\ k}$	Position of Hull Bottom Plating Pts along X_b (Ft)
$Y_{HULL\ k}$	Position of Hull Bottom Plating Pts along Y_b (Ft)
$Z_{HULL\ k}$	Position of Hull Bottom Plating Pts along Z_b (Ft)
$\eta_{SEA\ k}$	Height of Waves due to Seaway (Ft)
$\eta_{VGW\ k}$	Height of Waves due to Vehicle Generated Waves (Ft)
η_k	Total Height of Waves (Ft)
$H_{HULL\ k}$	Height of Hull Points above Mean Sea Level (Ft)
h_{wk}	Height of Hull Points above Water (Ft)

Cushion Compartments & Hull Bottom Points

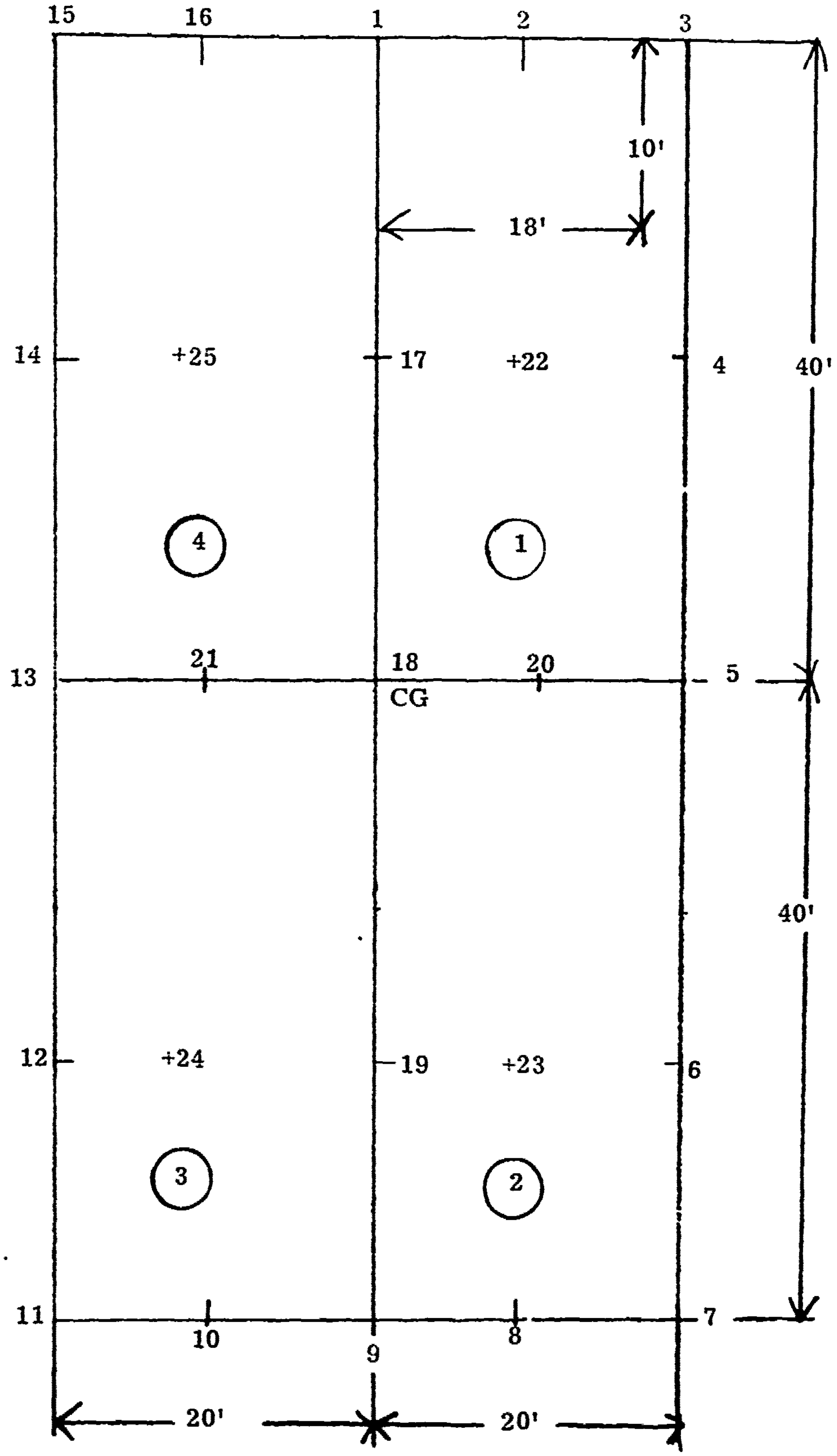


Figure 13. Hull Planform Profile (25 Pts)

TABLE 7. PLANFORM POINT LOCATIONS (25)

k	X_{HULL}	Y_{HULL}	Z_{HULL}	(Ft)
1	10	-18	12	
2	10	-8	12	
3	10	2	12	
4	-10	2	12	
5	-30	2	12	
6	-50	2	12	
7	-70	2	12	
8	-70	-8	12	
9	-70	-18	12	
10	-70	-28	12	
11	-70	-38	12	
12	-50	-38	12	
13	-30	-38	12	
14	-10	-38	12	
15	10	-38	12	
16	10	-28	12	
17	-10	-18	12	
18	-30	-18	12	
19	-50	-18	12	
20	-30	-8	12	
21	-30	-28	12	
22	-10	-8	12	
23	-50	-8	12	
24	-50	-28	12	
25	-10	-28	12	

CUSHION COMPARTMENT AIR VOLUMES. The average height of each cushion compartment hull bottom section is computed by a parabolic fit in both X and Y directions to the individual heights above water for that compartment. The air volume of each compartment is simply the product of this average height and the constant planform area.

Subroutine: VOLCUSH

Average Cushion Compartment Volumes

$$H_{CUSH\,1} = .0277778\,(h_{w1} + h_{w3} + h_{w5} + h_{w18}) + .1111111\,(h_{w2} + h_{w4} + h_{w20} + h_{w17}) + .4444444\,h_{w22}$$

$$H_{CUSH\,2} = .0277778\,(h_{w5} + h_{w7} + h_{w9} + h_{w18}) + .1111111(h_{w20} + h_{w6} + h_{w8} + h_{w19}) + .4444444\,h_{w23}$$

$$H_{CUSH\,3} = .0277778\,(h_{w18} + h_{w9} + h_{w11} + h_{w13}) + .1111111(h_{w21} + h_{w19} + h_{w10} + h_{w12}) + .4444444\,h_{w24}$$

$$H_{CUSH\,4} = .0277778\,(h_{w1} + h_{w18} + h_{w13} + h_{w15}) + .1111111(h_{w16} + h_{w17} + h_{w21} + h_{w14}) + .4444444\,h_{w25}$$

$$V_{CUSH\,n} = A_{CUSH}\,H_{CUSH\,n}$$

$n = 1----4$	Number of cushion compartments
$k = 1----25$	Number of hull bottom points
h_{wk}	Height of hull bottom points over water (Ft)
A_{CUSH}	Cushion compartment area = 800 Ft^2
$H_{CUSH\,n}$	Average height above water of cushion compartment (Ft)
$V_{CUSH\,n}$	Air volume of cushion compartment (Ft^3)

WAVE PUMPING. The wave pumping air flows are simply the rate of air volume chamge of each cushion compartment. In other words, wave pumping is the time derivative of each cushion volume and is calculated by a trapezoidal technique.

Subroutine: WV PUMP

$$Q_{PUMP\ n} = -\frac{d}{dt}\ V_{CUSH\ n}$$

$$= -\ \frac{V_{CUSH\ n_t} - V_{CUSH\ n_{t-\ \Delta t}}}{\Delta t}$$

$$= \frac{-\ 1.5\ V_{CUSH\ n_t} + 2.0\ V_{CUSH\ n_{t-\ \Delta t}} -\ .5\ V_{CUSH\ n_{t-2\Delta t}}}{\Delta t}$$

$n = 1---4$	Number of cushion compartments
$V_{CUSH\ n}$	Air Volume of Cushion Compartment (Ft^3)
$Q_{PUMP\ n}$	Rate of Compartment Volume Change (Ft^3/Sec)
t	Time (Sec)
Δ t	Sampling period (SEC)

CUSHION COMPARTMENT ESCAPE AREAS. The clearances between the bottom of the skirt and the water are calculated from the height over water of each hull point around the skirt periphery using a constant skirt height of 4.5 ft. These clearances are parabolically fit along the X, Y directions to obtain an average water clearance. This clearance is multiplied by the average cushion length (60 ft.) to get the escape area for each cushion compartment. However, the areas solved in the program are multiplied by a constant for later use.

Subroutine: SKAREA

Cushion Compartment Escape Area

$$CLR_k = h_{wk} - h_s \geq 0$$

$$AREA_1 = \frac{L}{6}\sqrt{\frac{2C_D}{\rho}}\left[CLR_1 + 4\ CLR_2 + 3\ CLR_3 + 8\ CLR_4 + 2\ CLR_5\right]$$

$$AREA_2 = \frac{L}{6}\sqrt{\frac{2C_D}{\rho}}\left[CLR_9 + 4\ CLR_8 + 3\ CLR_7 + 8\ CLR_6 + 2\ CLR_5\right]$$

$$AREA_3 = \frac{L}{6}\sqrt{\frac{2C_D}{\rho}}\left[CLR_9 + 4\ CLR_{10} + 3\ CLR_{11} + 8\ CLR_{12} + 2\ CLR_{13}\right]$$

$$AREA_4 = \frac{L}{6}\sqrt{\frac{2C_D}{\rho}}\left[CLR_1 + 4\ CLR_{16} + 3\ CLR_{15} + 8\ CLR_{14} + 2\ CLR_{13}\right]$$

k = 1----16	Number of cushion periphery points
h_{wk}	Height of hull bottom points over water (Ft)
h_s	Height of skirt = 4.5 Ft
CLR_k	Air gap between skirt and water at hull bottom points (Ft)
$AREA_1$	Cushion compartment 1 escape area (Ft^2). $\sqrt{\frac{2C_D}{\rho}}$
$AREA_2$	Cushion compartment 2 escape area (Ft^2). $\sqrt{\frac{2C_D}{\rho}}$
$AREA_3$	Cushion compartment 3 escape area (Ft^2). $\sqrt{\frac{2C_D}{\rho}}$
$AREA_4$	Cushion compartment 4 escape area (Ft^2). $\sqrt{\frac{2C_D}{\rho}}$
L	Width of Cushion Compartment = 20 Ft.
ρ	Density of Air = .00237 slugs/Ft^3
C_D	Discharge coefficient = .42

CUSHION AND MANIFOLD AIR PRESSURES. Six simultaneous equations are written for air flow between the 4 cushion compartments and the 2 manifolds and include the 4 wave pumping air flows. These air flow equations are rewritten by equating the sum of air flows with zero and by substituting the 6 pressure expressions in place of the air flows.

Because direct solution is not possible, a Newton-Raphson technique is developed. Setting the zero side of the equation to E, a set of error equations is defined. Taking the partial derivatives of each error with respect to each of the 6 pressures yields a 6 x 6 matrix. The change in pressure is then computed by inverting the 6 x 6 derivative matrix and multiplying by the error matrix. This change in pressure is simply added to the existing pressure. Iterations on this solution are necessary until the changes in pressure become very small.

Subroutine: CUSH P

Compartment Pressures

$$\overline{P} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} \begin{matrix} = \\ = \\ = \\ = \\ = \\ = \end{matrix} \begin{bmatrix} P_{CUSH\ 1} \\ P_{CUSH\ 2} \\ P_{CUSH\ 3} \\ P_{CUSH\ 4} \\ P_{MANS} \\ P_{MANP} \end{bmatrix}$$

$$\left[DEDP\right] = \left[DEDP\right]_{ij} = \frac{\partial E_i}{\partial P_j}$$

$$\Delta\overline{P} = \left[DEDP\right]^{-1} \overline{E}$$

$$\overline{P} = \overline{P} + \Delta\overline{P}$$

$$CRIT = \frac{\Delta\overline{P} \cdot \Delta\overline{P}}{\overline{P} \cdot \overline{P}}$$

If CRIT $\leq$ 0.0004 Solution Obtained, Otherwise Repeat Solution.

$$\begin{aligned}E_1 = {} & \text{QPUMP}_1 + 589\ \text{ASQRT}\ (P_5-P_1) + 675\ \text{ASQRT}\ (P_4-P_1)\\ & -338\ \text{ASQRT}\ (P_1-P_2) - \text{AREA}_1\ \text{ASQRT}(P_1)\\ & +1128\ C_{\text{SKIRT}}\text{ASQRT}\ (P_1)\ (109.0-P_1)\end{aligned}$$

$$\begin{aligned}E_2 = {} & \text{QPUMP}_2 + 589\ \text{ASQRT}\ (P_5-P_2) + 338\ \text{ASQRT}\ (P_1-P_2)\\ & -675\ \text{ASQRT}\ (P_2-P_3) - \text{AREA}_2\ \text{ASQRT}(P_2)\\ & +1128\ C_{\text{SKIRT}}\text{ASQRT}\ (P_2)\ (109.0 - P_2)\end{aligned}$$

$$\begin{aligned}E_3 = {} & \text{QPUMP}_3 + 589\ \text{ASQRT}\ (P_6-P_3) + 675\ \text{ASQRT}\ (P_2-P_3)\\ & -338\ \text{ASQRT}\ (P_3-P_4) -\text{AREA}_3\ \text{ASQRT}(P_3)\\ & +1128\ C_{\text{SKIRT}}\ \text{ASQRT}\ (P_3)\ (109.0 - P_3)\end{aligned}$$

$$\begin{aligned}E_4 = {} & \text{QPUMP}_4 + 589\ \text{ASQRT}\ (P_6-P_4) + 338\ \text{ASQRT}\ (P_3-P_4)\\ & -675\ \text{ASQRT}\ (P_4-P_1) -\text{AREA}_4\ \text{ASQRT}\ (P_4)\\ & +1128\ C_{\text{SKIRT}}\ \text{ASQRT}\ (P_4)\ (109.0 - P_4)\end{aligned}$$

$$\begin{aligned}E_5 = {} & -589\ \text{ASQRT}\ (P_5-P_1) -589\ \text{ASQRT}\ (P_5-P_2) -346\ \text{ASQRT}\ (P_5)\\ & - \frac{\text{NFANS}}{2000}\ (640\ \text{ASQRT}\ (P_5-300) + 15.8\ (P_5-300))\end{aligned}$$

$$\begin{aligned}E_6 = {} & -589\ \text{ASQRT}\ (P_6-P_3) -589\ \text{ASQRT}\ (P_6-P_4) -346\ \text{ASQRT}\ (P_6)\\ & - \frac{\text{NFAN P}}{2000}\ (640\ \text{ASQRT}\ (P_6-300) + 15.8\ (P_6-300))\end{aligned}$$

$$\begin{aligned}\partial E_1/\partial P_1 = {} & -589 \text{ DASQRT } (P_5-P_1) -675 \text{ DASQRT } (P_4-P_1)\\ & -338 \text{ DASQRT } (P_1-P_2) -\text{AREA}_1 \text{ DASQRT } (P_1)\\ & -1128\ C_{\text{SKIRT}} \text{ ASQRT } (P_1)\\ & +1128\ C_{\text{SKIRT}} \text{ DASQRT } (P_1)\ (109.0 - P_1)\end{aligned}$$

$$\partial E_1/\partial P_2 = 338 \text{ DASQRT } (P_1-P_2)$$

$$\partial E_1/\partial P_3 = 0$$

$$\partial E_1/\partial P_4 = 675 \text{ DASQRT } (P_4-P_1)$$

$$\partial E_1/\partial P_5 = 589 \text{ DASQRT } (P_5 -P_1)$$

$$\partial E_1/\partial P_6 = 0$$

$$\partial E_2/\partial P_1 = 338 \text{ DASQRT } (P_1-P_2) = \partial E_1/\partial P_2$$

$$\begin{aligned}\partial E_2/\partial P_2 = {} & -589 \text{ DASQRT } (P_5-P_2) -338 \text{ DASQRT } (P_1-P_2)-675 \text{ DASQRT}(P_2-P_3)\\ & -\text{AREA}_2 \text{ DASQRT } (P_2) -1128\ C_{\text{SKIRT}} \text{ ASQRT } (P_2)\\ & +1128\ C_{\text{SKIRT}} \text{ DASQRT } (P_2)\ (109.0 - P_2)\end{aligned}$$

$$\partial E_2/\partial P_3 = 675 \text{ DASQRT } (P_2-P_3)$$

$$\partial E_2/\partial P_4 = 0$$

$$\partial E_2/\partial P_5 = 589 \text{ DASQRT } (P_5 -P_2)$$

$$\partial E_2/\partial P_6 = 0$$

$$\partial E_3/\partial P_1 = 0$$

$$\partial E_3/\partial P_2 = 675\ \text{DASQRT}\ (P_2-P_3) = \partial E_2/\partial P_3$$

$$\begin{aligned}\partial E_3/\partial P_3 = & -589\ \text{DASQRT}\ (P_6-P_3)\ -675\ \text{DASQRT}\ (P_2-P_3)\\ & -338\ \text{DASQRT}\ (P_3-P_4)\ -\text{AREA}_3\text{DASQRT}\ (P_3)\\ & -1128\ C_{\text{SKIRT}}\ \text{ASQRT}\ (P_3)\ +1128\ C_{\text{SKIRT}}\text{DASQRT}(P_3)(109.0-P_3)\end{aligned}$$

$$\partial E_3/\partial P_4 = 338\ \text{DASQRT}\ (P_3-P_4)$$

$$\partial E_3/\partial P_5 = 0$$

$$\partial E_3/\partial P_6 = 589\ \text{DASQRT}\ (P_6-P_3)$$

$$\partial E_4/\partial P_1 = 675\ \text{DASQRT}\ (P_4-P_1) = \partial E_1/\partial P_4$$

$$\partial E_4/\partial P_2 = 0$$

$$\partial E_4/\partial P_3 = 338\ \text{DASQRT}\ (P_3-P_4) = \partial E_3/\partial P_4$$

$$\begin{aligned}\partial E_4/\partial P_4 = & -589\ \text{DASQRT}\ (P_6-P_4)\ -338\ \text{DASQRT}\ (P_3-P_4)\\ & -675\ \text{DASQRT}\ (P_4-P_1)\ -\text{AREA}_4\text{DASQRT}(P_4)\\ & -1128\ C_{\text{SKIRT}}\ \text{ASQRT}\ (P_4)\ +1128\ C_{\text{SKIRT}}\text{DASQRT}\ (P_4)(109.0-P_4)\end{aligned}$$

$$\partial E_4/\partial P_5 = 0$$

$$\partial E_4/\partial P_6 = 589\ \text{DASQRT}\ (P_6-P_4)$$

$$\partial E_5/\partial P_1 = 589 \text{ DASQRT } (P_5-P_1) = \partial E_1/\partial P_5$$

$$\partial E_5/\partial P_2 = 589 \text{ DASQRT } (P_5-P_2) = \partial E_2/\partial P_5$$

$$\partial E_5/\partial P_3 = 0$$

$$\partial E_5/\partial P_4 = 0$$

$$\partial E_5/\partial P_5 = -589 \text{ DASQRT } (P_5-P_1) -589 \text{ DASQRT } (P_5-P_2) -346 \text{ DASQRT}(P_5) - \frac{\text{NFANS}}{2000} .(640 \text{ DASQRT } (P_5-300) + 15.8)$$

$$\partial E_5/\partial P_6 = 0$$

$$\partial E_6/\partial P_1 = 0$$

$$\partial E_6/\partial P_2 = 0$$

$$\partial E_6/\partial P_3 = 589 \text{ DASQRT } (P_6-P_3) = \partial E_3/\partial P_6$$

$$\partial E_6/\partial P_4 = 589 \text{ DASQRT } (P_6-P_4) = \partial E_4/\partial P_6$$

$$\partial E_6/\partial P_5 = 0$$

$$\partial E_6/\partial P_6 = -589 \text{ DASQRT } (P_6-P_3) -589 \text{ DASQRT } (P_6-P_4) -346 \text{ DASQRT } (P_6) - \frac{\text{NFAN P}}{2000} (640 \text{ DASQRT } (P_6-300)+15.8)$$

$$[\mathrm{DEDP}]_{ij} = \begin{bmatrix} \partial E_1/\partial P_1 & \partial E_1/\partial P_2 & \partial E_1/\partial P_3 & \partial E_1/\partial P_4 & \partial E_1/\partial P_5 & \partial E_1/\partial P_6 \\ \partial E_2/\partial P_1 & \partial E_2/\partial P_2 & \partial E_2/\partial P_3 & \partial E_2/\partial P_4 & \partial E_2/\partial P_5 & \partial E_2/\partial P_6 \\ \partial E_3/\partial P_1 & \partial E_3/\partial P_2 & \partial E_3/\partial P_3 & \partial E_3/\partial P_4 & \partial E_3/\partial P_5 & \partial E_3/\partial P_6 \\ \partial E_4/\partial P_1 & \partial E_4/\partial P_2 & \partial E_4/\partial P_3 & \partial E_4/\partial P_4 & \partial E_4/\partial P_5 & \partial E_4/\partial P_6 \\ \partial E_5/\partial P_1 & \partial E_5/\partial P_2 & \partial E_5/\partial P_3 & \partial E_5/\partial P_4 & \partial E_5/\partial P_5 & \partial E_5/\partial P_6 \\ \partial E_6/\partial P_1 & \partial E_6/\partial P_2 & \partial E_6/\partial P_3 & \partial E_6/\partial P_4 & \partial E_6/\partial P_5 & \partial E_6/\partial P_6 \end{bmatrix}$$

$$= \begin{bmatrix} \partial E_1/\partial P_1 & \partial E_1/\partial P_2 & 0 & \partial E_1/\partial P_4 & \partial E_1/\partial P_5 & 0 \\ \partial E_1/\partial P_2 & \partial E_2/\partial P_2 & \partial E_2/\partial P_3 & 0 & \partial E_2/\partial P_5 & 0 \\ 0 & \partial E_2/\partial P_3 & \partial E_3/\partial P_3 & \partial E_3/\partial P_4 & 0 & \partial E_3/\partial P_6 \\ \partial E_1/\partial P_4 & 0 & \partial E_3/\partial P_4 & \partial E_4/\partial P_4 & 0 & \partial E_4/\partial P_6 \\ \partial E_1/\partial P_5 & \partial E_2/\partial P_5 & 0 & 0 & \partial E_5/\partial P_5 & 0 \\ 0 & 0 & \partial E_3/\partial P_6 & \partial E_4/\partial P_6 & 0 & \partial E_6/\partial P_6 \end{bmatrix}$$

ASQRT Y= ASQRT (X)

$$y = \sqrt{|x|}\,\frac{|x|}{x} \qquad |x| > 1$$

$$y = x \qquad |x| \leq 1$$

DASQRT Y = DASQRT (X)

$$y = \frac{1}{2\sqrt{|x|}} \qquad |x| > 1$$

$$y = 1 \qquad |x| \leq 1$$

$P_{CUSH\ 1}$	Pressure in cushion compartment 1 (Lbs/Ft^2)
$P_{CUSH\ 2}$	Pressure in cushion compartment 2 (Lbs/ Ft^2)
$P_{CUSH\ 3}$	Pressure in cushion compartment 3 (Lbs/ Ft^2)
$P_{CUSH\ 4}$	Pressure in cushion compartment 4 (Lbs/ Ft^2)
P_{MANS}	Pressure in STBD manifold (Lbs/ Ft^2)
$P_{MAN\ P}$	Pressure in PORT manifold (Lbs/Ft^2)
$\overline{E}$	Compartment air flow errors; ideally = 0
$\Delta\overline{P}$	Change in 6 compartment pressures per iteration (Lbs/Ft^2)
C_{SKIRT}	Skirt rigidity coefficient = .01
$Q_{PUMP\ 1}$	Rate of cushion compartment 1 volume change (Ft^3/ Sec)
$Q_{PUMP\ 2}$	Rate of cushion compartment 2 volume change (Ft^3/ Sec)
$Q_{PUMP\ 3}$	Rate of cushion compartment 3 volume change (Ft^3/Sec)
$Q_{PUMP\ 4}$	Rate of cushion compartment 4 volume change (Ft^3/ Sec)
N_{FANS}	Angular speed of STBD fan (RPM)
$N_{FAN\ P}$	Angular speed of PORT fan (RPM)
$AREA_1$	Cushion Compartment 1 escape area (Ft^2) • Constant
$AREA_2$	Cushion Compartment 2 escape area (Ft^2) • Constant
$AREA_3$	Cushion Compartment 3 escape area (Ft^2) • Constant
$AREA_4$	Cushion Compartment 4 escape area (Ft^2) • Constant

ENVIRONMENT

The environmental effects on the vehicle consist of only wind and waves.

WIND. The implementation of wind consists only of constant speed and constant direction effects. The wind velocity is combined with vehicle velocity to yield the apparent head and side wind velocities. Also, the total apparent wind velocity and angle of attack are determined.

Subroutine: WIND PAR

WIND

$$u_{WIND} = u + S_{WIND} \cos (\psi - \psi_{WIND})$$

$$v_{WIND} = v - S_{WIND} \sin (\psi - \psi_{WIND})$$

$$V^2_{AWIND} = u^2_{WIND} + v^2_{WIND}$$

$$V_{AWIND} = \sqrt{V^2_{AWIND}}$$

$$\beta_a = \tan^{-1} \frac{v_{WIND}}{u_{WIND}}$$

S_{WIND}	Wind Velocity (Ft/Sec)
ψ_{WIND}	Set of Wind (DEG)
ψ	Heading of Vehicle (DEG)
u	Velocity (Linear) in direction X_b (Ft/Sec)
v	Velocity (Linear) in direction Y_b (Ft/Sec)
u_{WIND}	Apparent Head Wind Velocity along X_b (Ft/Sec)
v_{WIND}	Apparent Side Wind Velocity along Y_b (Ft/Sec)
V_{AWIND}	Apparent Wind Velocity (Ft/Sec)
β_a	Apparent Wind Angle ($\pm 180^o$)

SEAWAY. These waves are generated as a single component sinusoid moving northward. These waves are meant to simulate swell consisting of relatively long period and wave length. Computation limits in real-time prohibit the addition of other frequency components.

Because this vehicle is meant to be a landing craft, considerable effort in this dynamic seaway simulation is devoted to the creation of shoaling and breaking waves. Shoaling waves naturally blend into a single frequency making a realistic offshore many component wave generation superfluous. As the wave approaches the beach, which runs east-west, the wave length shortens and the wave amplitude increases causing the wave slope to steepen. Once the wave slope becomes sufficiently steep, the wave breaks causing the amplitude to decrease sharply. Then, as the wave continues to encroach upon the beach, the amplitude continues to decrease until it becomes zero at the boundary of the beach.

The waves are created completely in the earth-fixed coordinate system. It is necessary to obtain the wave heights at all the hull bottom plating points for use in calculating forces and moments. Consequently, it is first necessary to calculate the inertial components of all hull points and the earth-fixed position of each hull point.

Over land there is no seaway, so for the particular hull point that is over land, its associated wave height is set to zero.

Over the deep ocean region (offshore) a simple cosine-function is evaluated using a phase angle based on position and the wave frequency.

Over the shoaling region, the computational problem begins to get more difficult. From the depth of water directly beneath each hull point, the wave number and phase angle at that point must be determined. From the water depth, wave number, and offshore wave number and amplitude, the actual wave amplitude under each hull point is calculated and limited. The wave height of the first order is a cosine function of the local amplitude and phase and offshore wave frequency. The second order wave height is a cosine function of twice the local phase and offshore wave frequency multiplied by a polynomial in offshore and local wave numbers and multiplied by the square of the local wave amplitude. The final wave height under each hull bottom plating point is the sum of the first and second order wave heights.

Figure 14 defines the ocean boundary regions in the horizontal plane relative to the earth-fixed coordinate system. Figure 15 defines the offshore parameters and relates the shoaling region parameters geometrically in the vertical plane.

The wave number and phase angle functions are piecewise fit ($f(x)=A_i/x+B_ix+C_i$) of 9 equal segments and 2 half segments using a least squared fit and adjusted for function continuity at the segment boundaries. See Figures 16 and 17 for computer generated plots of these functions in the shoaling region.

Figure 18 shows a wave profile (snapshot in time versus position) as it travels from the deep ocean onto the beach. Figure 19 shows an expanded section at the beach of Figure 18.

Subroutine: CVIHULL

$$\left\{\begin{matrix} X_{iHk} \\ Y_{iHk} \\ Z_{iHk} \end{matrix}\right\} = C_i^v \left\{\begin{matrix} X_{HULL\ k} \\ Y_{HULL\ k} \\ Z_{HULL\ k} \end{matrix}\right\}$$

$$= \left\{\begin{matrix} X_{HULL\ k} \operatorname{Cos} \psi - Y_{HULL\ k} \sin \psi \\ X_{HULL\ k} \operatorname{Sin} \psi + Y_{HULL\ k} \cos \psi \\ -X_{HULL\ k}\ \theta + Y_{HULL\ k}\ \phi + Z_{HULL\ k} \end{matrix}\right\}$$

k	Number of Hull Bottom Plating Points
C_i^v	Vehicle to Inertial Transformation
$X_{HULL\ k}$	Position of Hull Bottom Plating Pts. along X_b (Ft)
$Y_{HULL\ k}$	Position of Hull Bottom Plating Pts. along Y_b (Ft)
$Z_{HULL\ k}$	Position of Hull Bottom Plating Pts. along Z_b (Ft)
ϕ	Euler Angle Roll (RAD)
θ	Euler Angle Pitch (RAD)
ψ	Heading of Vehicle (DEG)
X_{iHk}	Components of Hull Bottom Points along X_i (Ft)
Y_{iHk}	Components of Hull Bottom Points along Y_i (Ft)
Z_{iHk}	Components of Hull Bottom Points along Z_i (Ft)

Dynamic Seaway Generation

Subroutine: SEAWAY

Initial Calculations

$$\omega_o = \frac{2\pi}{\tau}$$

$$\lambda_o = \frac{g}{2\pi}\tau^2$$

$$D_s = \frac{\lambda_o}{2}$$

$$K_o = \frac{\omega_o^2}{g}$$

$$X_s = \frac{-D_s}{m_B}$$

Subroutine: OCEAN

$$\begin{Bmatrix} X_{i\ HULL\ k} \\ Y_{i\ HULL\ k} \\ Z_{i\ HULL\ k} \end{Bmatrix} = \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} + \begin{Bmatrix} X_{iHk} \\ Y_{iHk} \\ Z_{iHk} \end{Bmatrix}$$

Over Land $\quad X_{i\,HULL\,k} \geq 0$

$\eta_{SEA\,k} \quad 0$

Over Deep Ocean Region $\quad X_{i\,HULL\,k} \leq X_s$

$$\eta_{SEA\,k} = \frac{H}{2} \cos \left[K_o \left(X_{i\,HULL\,k} - X_s \right) - \omega_o t \right]$$

Over Shoaling Region $\quad X_{i\,HULL\,k} > X_s$

$$D_x = - m_B X_{i\,HULL\,k}$$

$$K_x : K_o = K_x \tanh K_x D_x ; \; K_x = \frac{f(D_x/D_s)}{\tau^2}$$

$$\nu_x = \int_{D_s}^{0} K_x \, dD_x \approx \frac{f(D_x/D_s)(-X_s)}{\tau^2}$$

$$H_x - H \left[\frac{K_x}{K_x^2 D_x - K_o^2 D_x + K_o} \right]^{1/2}$$

$$H_{x_{max}} = .78 \, D_x$$

$$H_{x_{max}} = .142 \, \frac{2\pi}{K_x}$$

$$\eta_1 = \frac{H_x}{2} \cos(\gamma_x - \omega_o t)$$

$$2\eta_2 = H_x^2 \left[\frac{3}{8} \frac{K_x^4}{K_o^3} - \frac{7}{8} \frac{K_x^2}{K_o} - \frac{1}{8} \frac{K_o^3}{K_x^2} + \frac{5}{8} K_o \right] \leq \frac{H_x}{4}$$

$$\eta_{SEA_k} = \eta_1 + \eta_2 \cos 2(\gamma_x - \omega_o t)$$

$$= \eta_1 + 2\eta_2 \left[\cos^2(\gamma_x - \omega_o t) - 1/2 \right]$$

k	Number of Hull Bottom Plating Points
λ_o	OFFSHORE Wave Length (Ft)
τ	OFFSHORE Wave Period (SEC)
ω_o	OFFSHORE Wave Frequency (RAD/SEC)
g	Gravitational Acceleration = 32.17 ft/sec^2
D_s	Depth at Which Shoaling Begins (Ft)
K_o	OFFSHORE Wave Number (Rad/Ft)
H	Peak to Peak Wave Height (Ft)
m_B	Slope of the Ocean Bottom
X_S	Position at which Shoaling begins along X_i (Ft)
X_i	Inertial Position (North) of Pilot Cabin (Ft)
Y_i	Inertial Position (East) of Pilot Cabin (Ft)
Z_i	Inertial Position (Vertical) of Pilot Cabin (Ft)
X_{iHk}	Components of hull bottom points along X_i (Ft)
Y_{iHk}	Components of hull bottom points along Y_i (Ft)
Z_{iHk}	Components of hull bottom points along Z_i (Ft)
$X_{iHULL\ k}$	Position of hull bottom plating pts. along X_i (Ft)
$Y_{iHULL\ k}$	Position of Hull bottom plating pts. along Y_i (Ft)
$Z_{iHULL\ k}$	Position of hull bottom plating pts. along Z_i (Ft)
D_x	Ocean depth under hull point along Z_i (Ft)
K_x	Wave number at hull point (Rad/Ft)

$\surd_x$	Wave phase angle at hull point (Rad)
η_1	First Order Wave Height at Pts. under Hull (Ft)
η_2	Stokes Second Order Correction (Ft)
$\eta_{sea\ k}$	Height of Waves due to Seaway (Ft)
t	Time (Sec)

Subroutine: SYSPRINT

$$C = \frac{\omega_o}{K_x}$$

$$T_e = \frac{2\pi}{\omega_e} = \frac{\lambda_x}{\dot{X}_i - C}$$

$$\omega_e = K_x \dot{X}_i - \omega_o$$

ω_o	Offshore wave frequency (Rad/Sec)
K_x	Wave number at pilot station (Rad/Ft)
C	Wave velocity along X_i (Ft/Sec)
λ_x	Wave length at pilot station (Ft)
$\dot{X}_i$	Vehicle velocity along X_i (Ft/Sec)
ω_e	Wave encounter frequency (Rad/Sec)

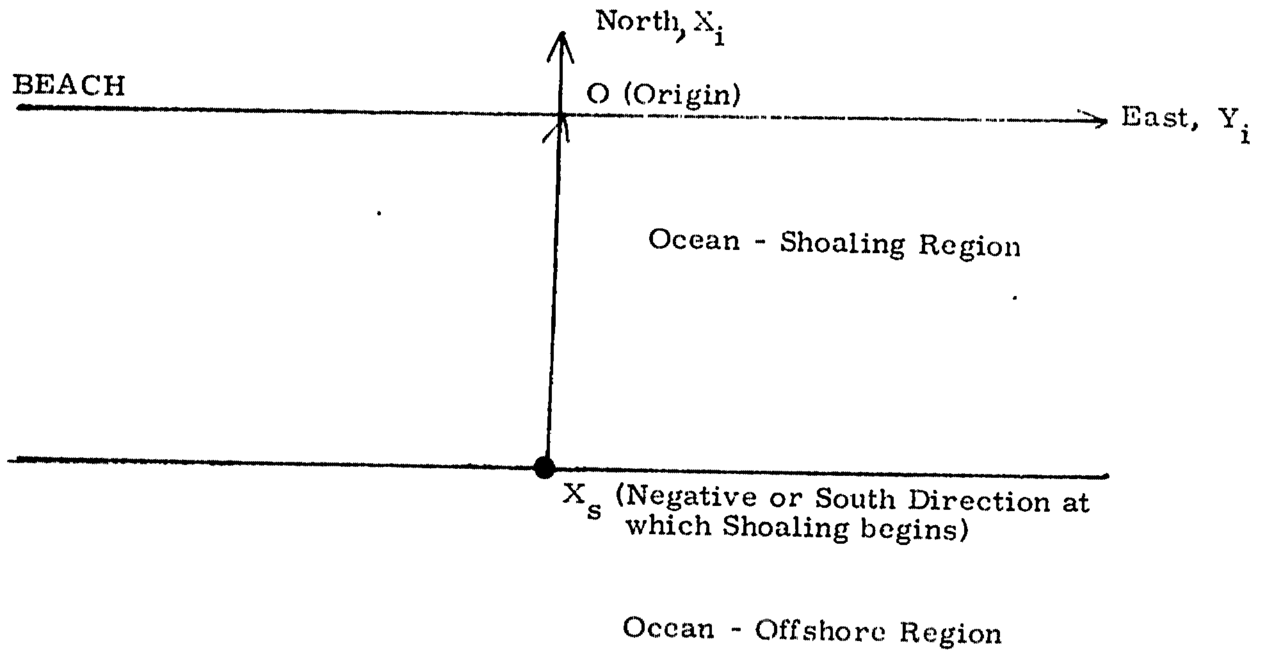
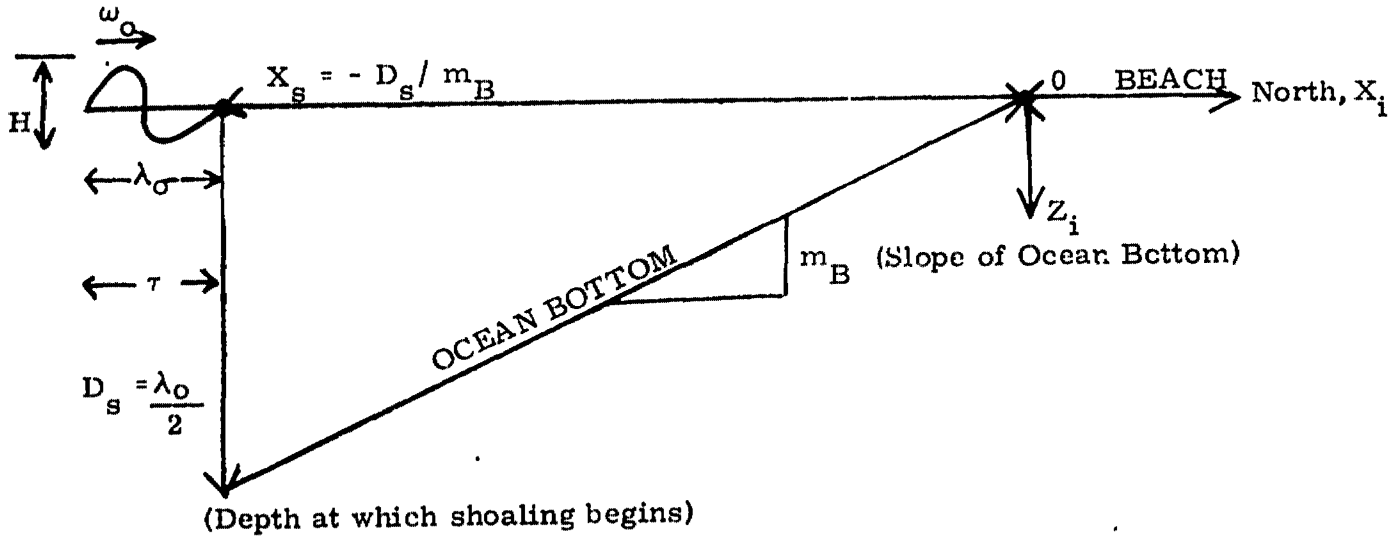


Figure 14. Ocean-Land Definition


ω_o
H
$X_s = - D_s / m_B$
0
BEACH
North, X_i
λ_o
τ
Z_i
m_B (Slope of Ocean Bottom)
OCEAN BOTTOM
$D_s = \frac{\lambda_o}{2}$
(Depth at which shoaling begins)

H	Wave height at crossover (0 to 4 nominal, 8 extreme) (Ft)
τ	Period of wave at crossover (Sec)
λ_o	Wave length at crossover (Ft)
ω_o	Offshore wave frequency (Rad/Sec)

Figure 15. Ocean Bottom Profile

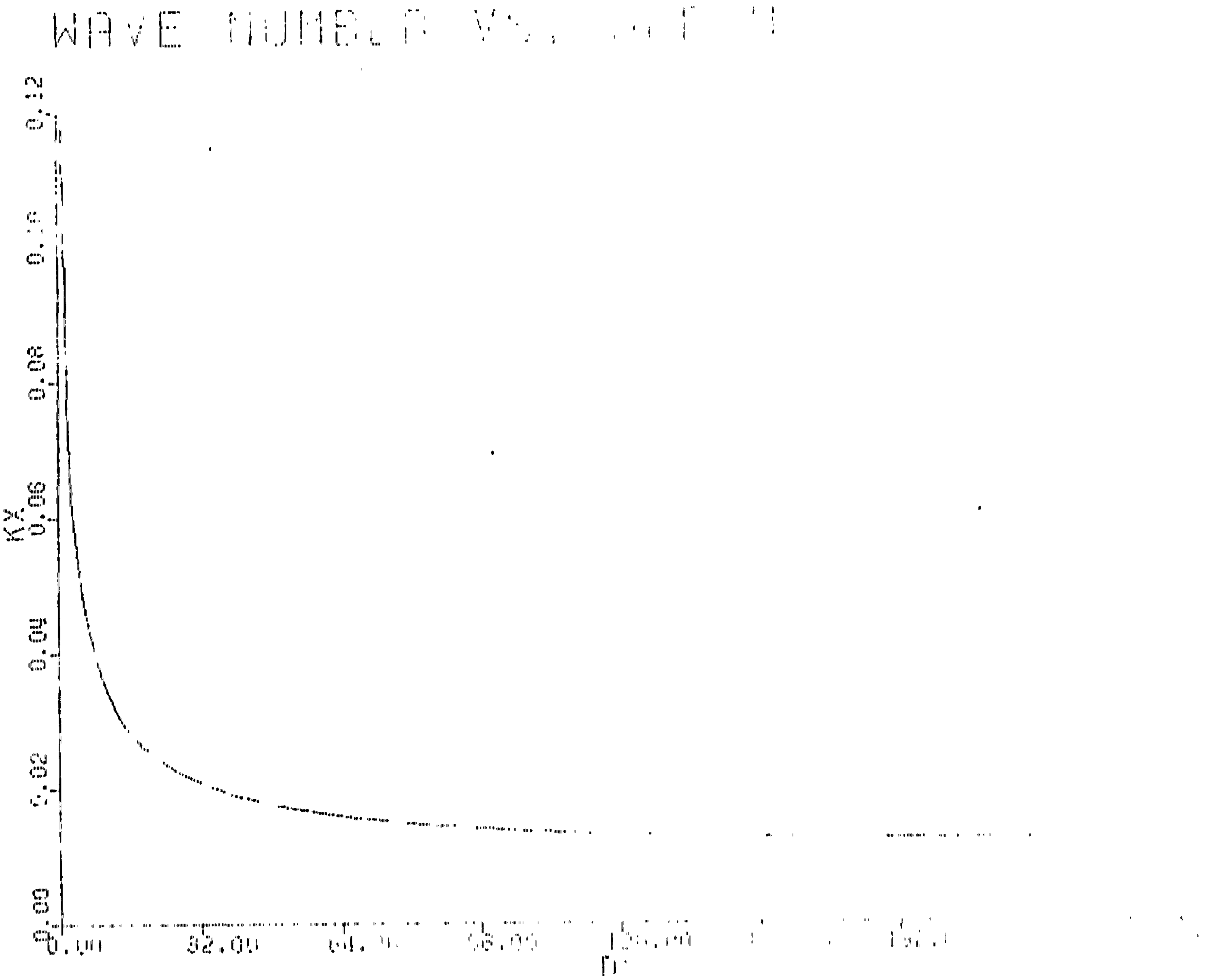


Figure 16. Wave Number

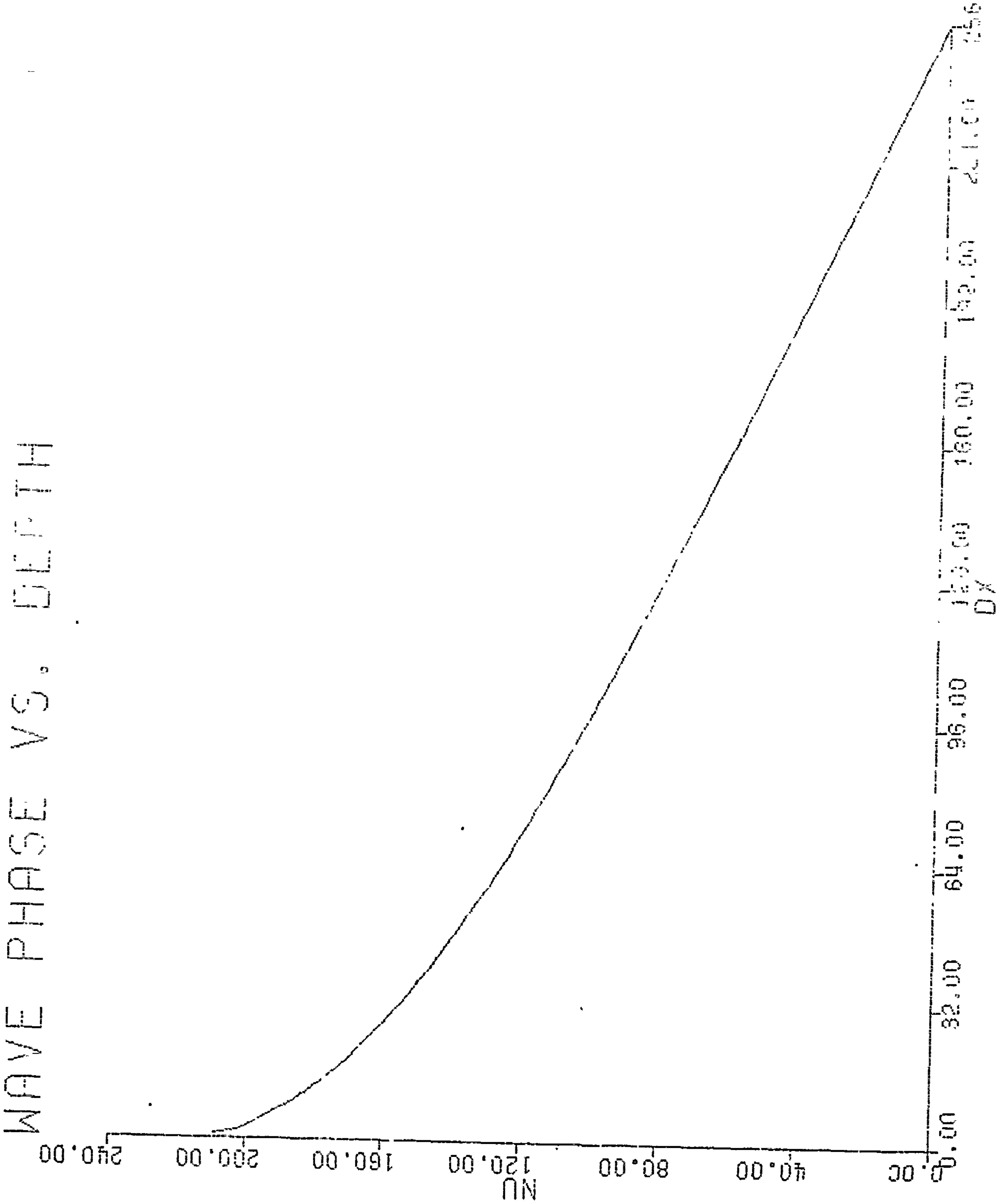
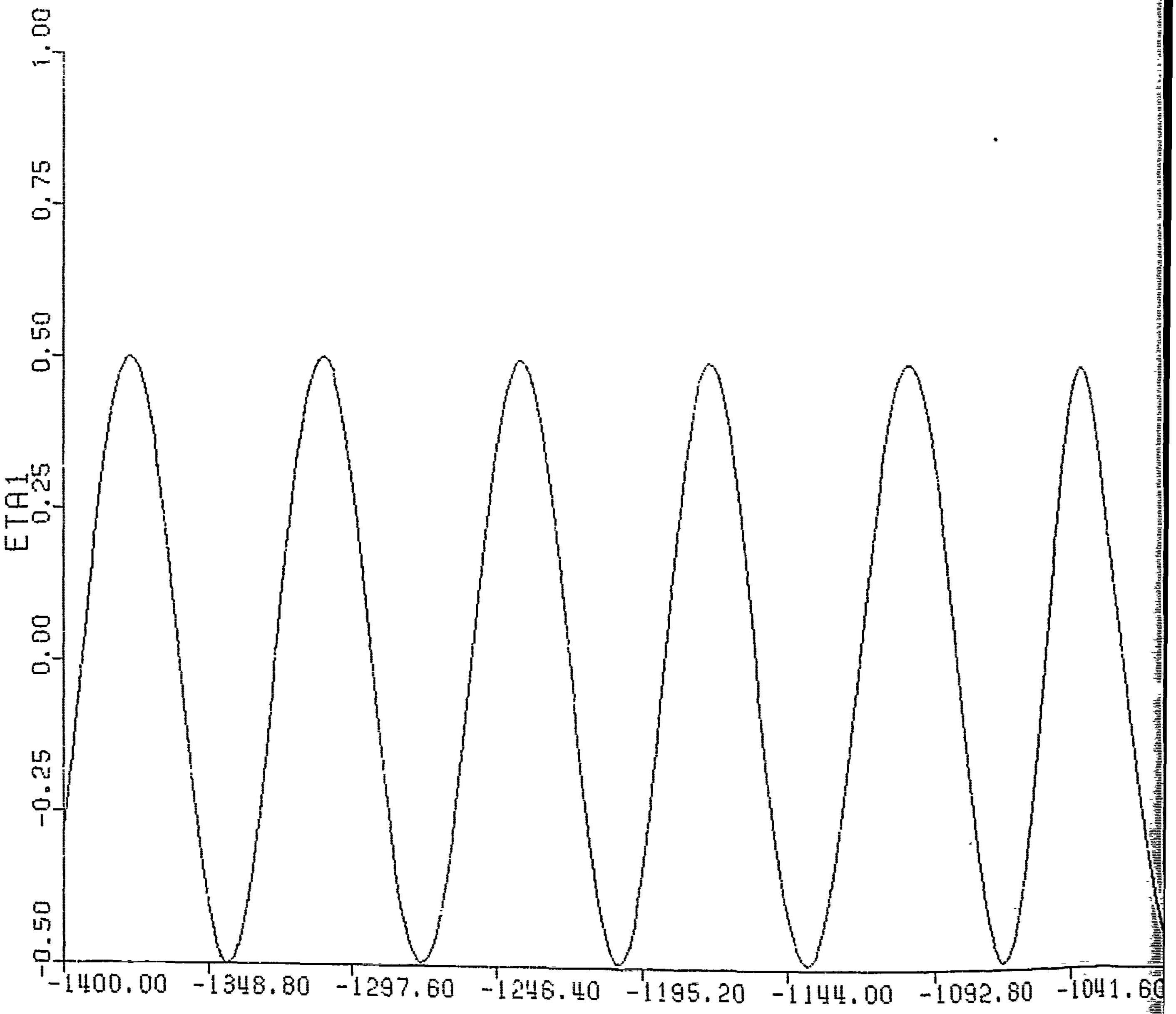
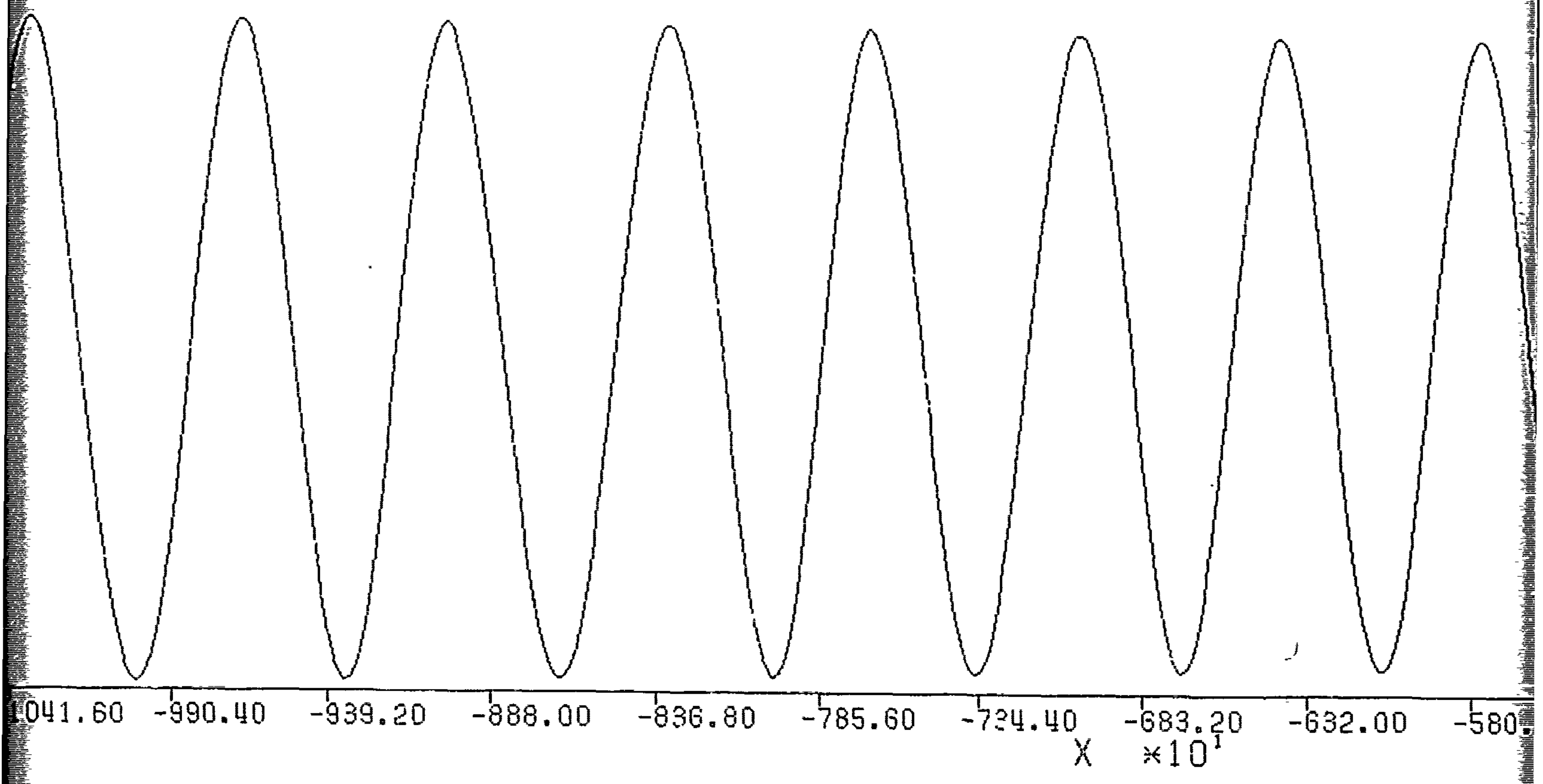
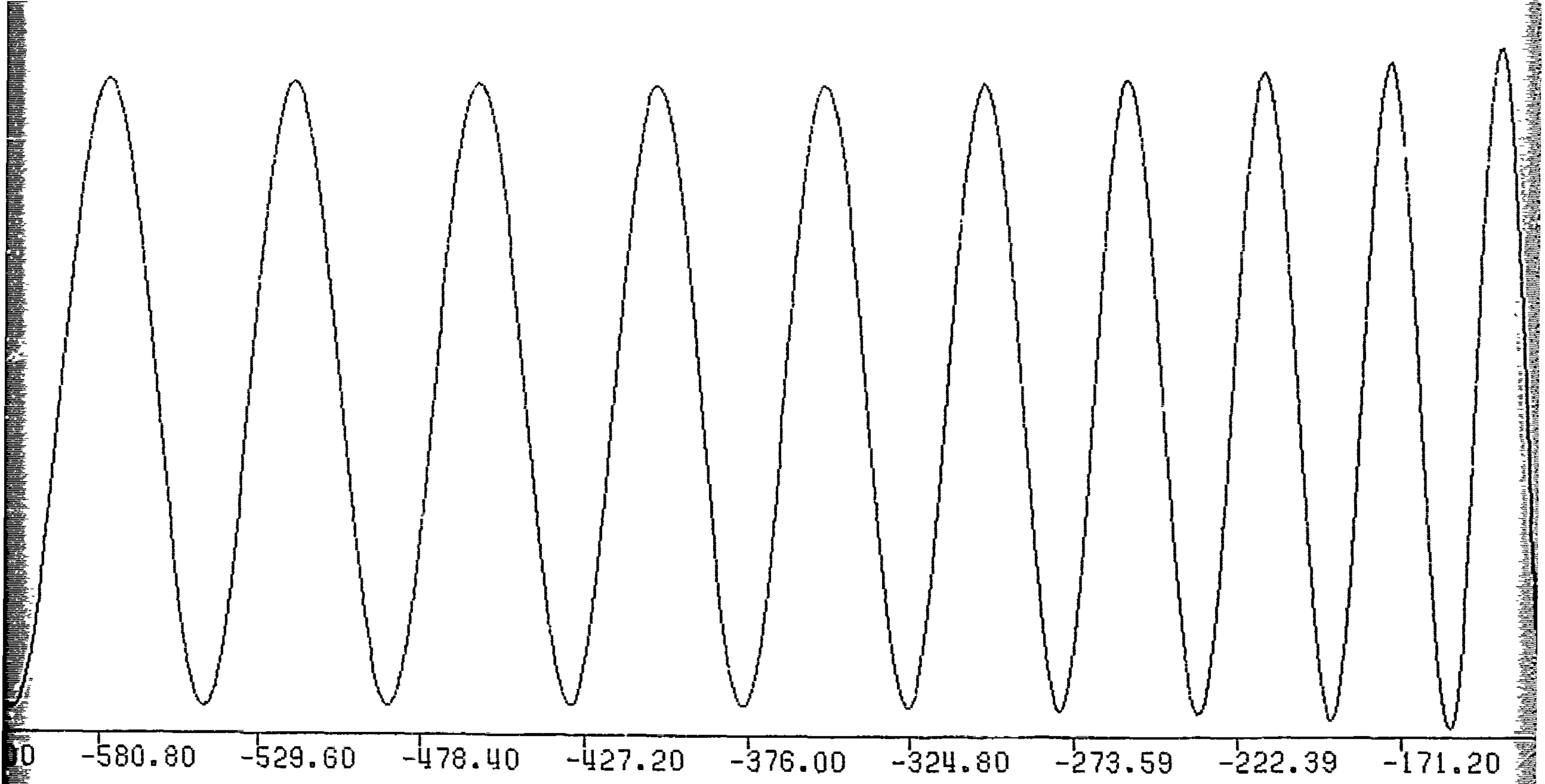


Figure 17. Wave Phase


ETA1
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-1400.00
-1348.80
-1297.60
-1246.40
-1195.20
-1144.00
-1092.80
-1041.60


1041.60
-990.40
-939.20
-888.00
-836.80
-785.60
-734.40
-683.20
-632.00
-580.
X *10^1

2



Figure

3

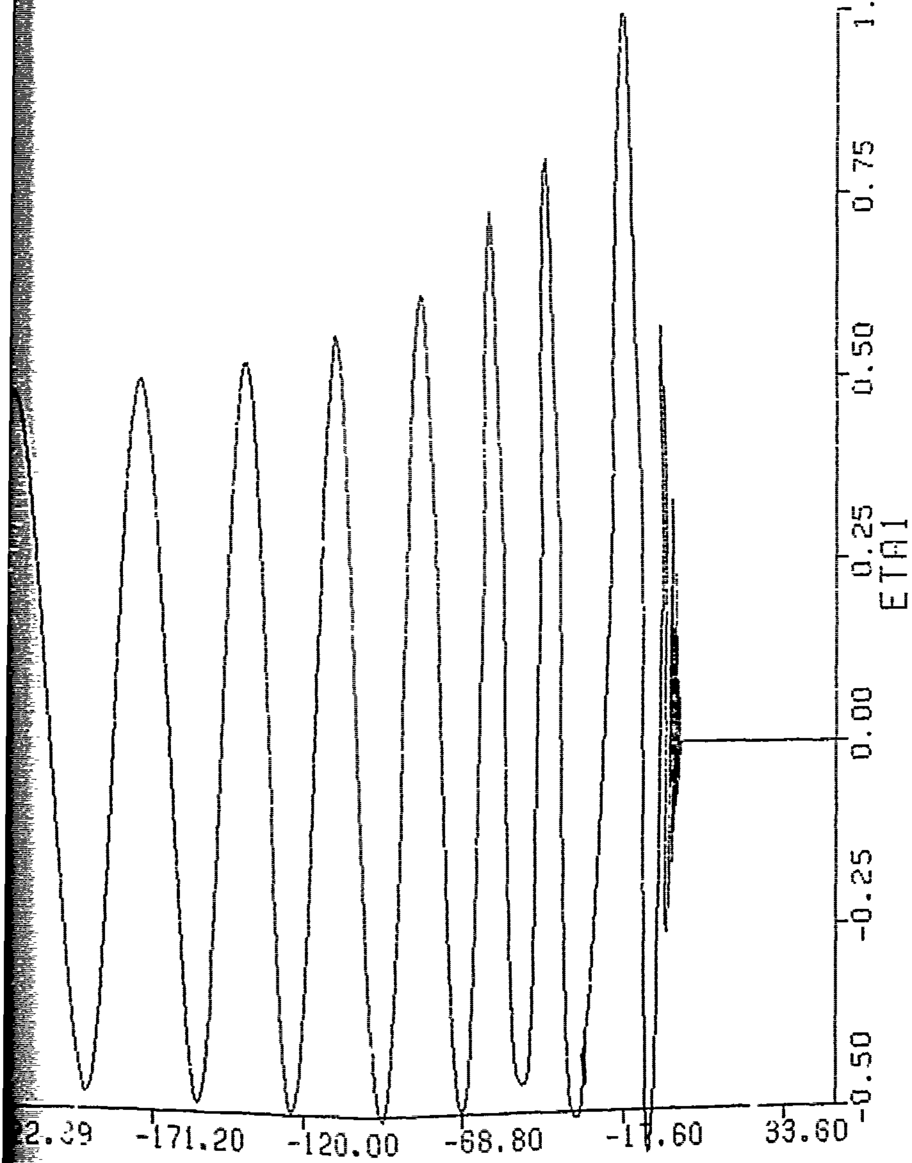


Figure 18. Ocean Wave Profile

4

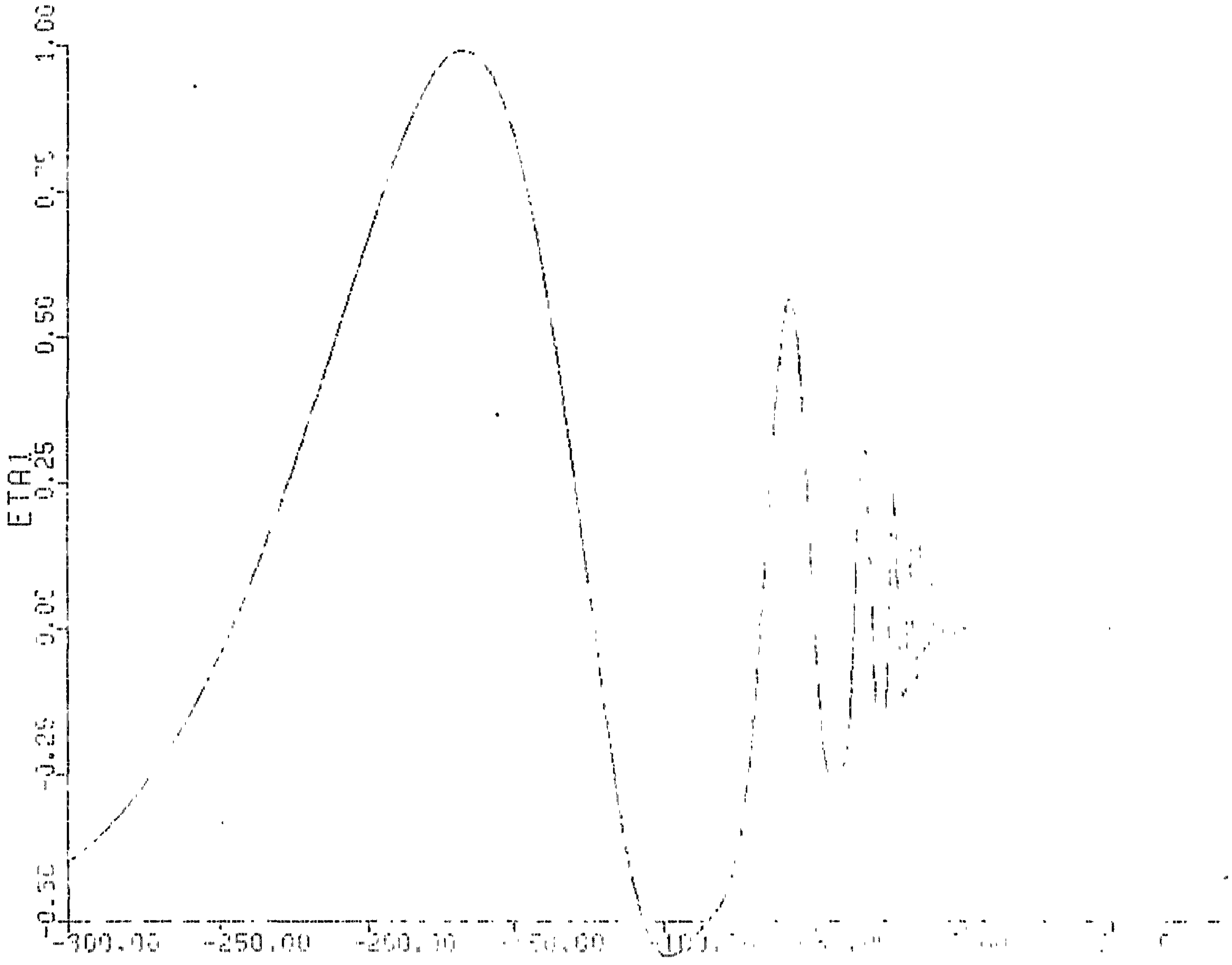


Figure 19. Ocean Wave at Beach

VEHICLE GENERATED WAVES

This section of the mathematical model is the most complex part of the whole simulation. The implementation and the concept are reasonably straight forward, but the mathematics of the kernel function and the development of the model are sophisticated and complex. Although the implementation is straightforward, the number of calculations is voluminous. This fact presents great difficulty in doing these calculations in real-time.

The vehicle rides on a cushion of air which is obviously pressurized to counteract the force of gravity. The higher-than-atmospheric pressure causes a depression in the water surface. If the craft is not moving, i.e., just hovering, this surface depression is static in nature. This situation can be explained as a sequence of pressure impulses applied on the free surfaces for a prolonged period of time. If the duration over which these impulses are applied is short, a small number of radiating waves would be generated. In the limit, however, the shape under the cushion would be static in shape. The present implementation only allows for a finite length of time for application of impulses (T_O = 8 sec). Hence the "static" free-surface elevation will still be wavy outside of the air cushion. It is this wave pattern which constitutes the vehicle generated waves. No concern is given to the influence of these waves beyond the confines of the hull.

After a wave pattern is created and the vehicle moves, its forward motion will create another wave pattern. As the vehicle moves over the established wave pattern, distortion of the existing wave pattern occurs under the vehicle because of the newly generated pattern. This changing of wave shape from one pattern to the next mathematically can be expressed as a convolution. Therefore, to create a complete profile of the vehicle generated wave pattern, it is necessary to convolve all wave shapes generated in the past with the present.

The pressure distribution under the craft used for rigid body dynamics is dependent upon the air volume and leakage gaps across the complete planform. In order to know the complete distribution of the air volume, it is necessary to determine the height between planform and water at all points under the hull. This is synonymous with knowing the height of the wave at each point with respect to mean water level.

The model of the vehicle generated waves is only concerned with determining these wave heights by assuming a time-independent pressure distribution. The change in pressure and air volume distributions are considered elsewhere using the vehicle generated wave heights.

WAVE HEIGHTS. The vehicle generated wave heights are determined under a selective number of points of the hull planform. This determination is made by performing a convolution integral over the kernel function at specific time intervals in the past.

The kernel function is a three dimensional wave height function of position (X, Y) and time. This function is pretabulated versus X and Y for specific sampling periods past (t-nΔt). In other words, a wave pattern is established at each time period in the past by table lookup as a function of position (how far the vehicle has travelled since being at the previous point). These kernels are then convolved over all time yielding the wave height.

Over land there is no water to be displaced, the associated VGW height is set to zero. If the complete vehicleis over land, the skirt and spray drag coefficient is set to zero.

The history of the trajectory is stored and updated at each time step. This trajectory history consists of the change in inertial position of the vehicle from the present and the heading of the vehicle at each time step in the past. The change in inertial position of each hull point is calculated from the vehicle change in position and the inertial hull point component. This change in position is transformed from earth-fixed coordinates to body-fixed coordinates for each time period. This change in position is then referenced to the vehicle center of gravity. Because the VGW is symmetrical about the body center, only the magnitude of this change in X, Y position is needed to arrive at the kernel value. All kernel values are integrated by Simpson's Rule and dimensionalized in time and position to yield the final VGW height under each planform point.

Complete plots were generated to show the physical shape of the Vehicle Generated Wave (VGW). These plots show the wave for vehicle straight line motion and constant forward speed for all time. Three sets of plots are included for vehicle forward speed of 0, 30, and 84 ft/sec corresponding to 0, 18, and 50 knots. Each set consists of three plots. The first plot shows the VGW height (η) along the longitudinal direction of craft motion. The second plot shows the VGW height (η) along the lateral direction of the vehicle. The third plot shows the VGW height (η) in time at a fixed point on the ocean surface as the vehicle passes over it.

The first two wave profiles (see Figures 20, 21, 23, 24, 26, and 27) are [illegible] dimensional cuts (slices) along the vehicle body axes (x, y) centered about the CG cb snapshotted at a particular instant in time; although, the wave profile as seen from the vehicle does not change when its speed is constant. The first wave profile (see Figures 20, 23, 26) versus x position shows the CG at x=0 with the bow at x=40 and the stern at x=-40. The second wave profile (see Figures 21, 24, 27) versus y position shows the CG at Y=0 with the starboard side at y=20 and the port side at y=20.

The third wave profile (see Figures 22, 25, 28) shows the two dimensional wave height effect of a fixed point on the ocean surface as the centerline of the ACV approaches and passes over the point. At t=0 the CG is directly over the point.

Subroutine: VG WAVE

Over Land

$$\eta_{VGW\ k} = 0$$

$$C_{D\ SKIRT} = 0$$

Over Ocean

$$C_{D\ SKIRT} = +.25$$

$$\begin{Bmatrix} \Delta X_{iH} \\ \Delta Y_{iH} \\ \Delta Z_{iH} \end{Bmatrix}_n = \begin{Bmatrix} X_{iH} \\ Y_{iH} \\ Z_{iH} \end{Bmatrix}_k + \begin{Bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \end{Bmatrix}_n$$

$$\begin{Bmatrix} \Delta X_{vH} \\ \Delta Y_{vH} \\ \Delta Z_{vH} \end{Bmatrix} = C^{i}_{v_n} \begin{Bmatrix} \Delta X_{iH} \\ \Delta Y_{iH} \\ \Delta Z_{iH} \end{Bmatrix}_n$$

$$\begin{Bmatrix} R_x \\ R_y \\ R_z \end{Bmatrix}_n = \left| \begin{Bmatrix} \Delta X_{vH} \\ \Delta Y_{vH} \\ \Delta Z_{vH} \end{Bmatrix}_n - \begin{Bmatrix} X_{CG} \\ Y_{CG} \\ Z_{CG} \end{Bmatrix} \right|$$

$$\eta_{VGW_k} = \frac{Po}{\rho_w g} \sqrt{\frac{g}{a}} \int_{t-N\Delta t}^{t} K_{xy} (R_{x_n}, R_{y_n}, \tau_n)\, d\tau$$

$$\Delta X_{iH_n} = X_{iH_k} + \Delta X_{i_n}$$

$$\Delta Y_{iH_n} = Y_{iH_k} + \Delta Y_{i_n}$$

$$\Delta X_{vH_n} = \Delta X_{iH_n} \text{Cos } \psi_n + \Delta Y_{iH_n} \text{Sin } \psi_n$$

$$\Delta Y_{vH_n} = -\Delta X_{iH_n} \text{Sin } \psi_n + \Delta Y_{iH_n} \text{Cos } \psi_n$$

$$R_{xn} = \left| \Delta X v_{H_n} - X_{CG} \right|$$

$$R_{yn} = \left| \Delta Y v_{H_n} - Y_{CG} \right|$$

$$\eta_{VGW\,k} = -1.5399172 \frac{\Delta t}{1.5} \left[K_{xy_2} + K_{xy_N} + 1.5\, K_{xy_4} + 4 \sum_{i=5,7,9\ldots}^{N-1} K_{xy\,i} + 2 \left(K_{xy_1} + K_{xy_3} + \sum_{i=6,8,10\ldots}^{N-2} K_{xyi} \right) \right]$$

k	Number of Hull Bottom Plating Points
$n = 1, \ldots, N$	Number of History states, N is an even positive integer
$\eta_{VGW\ k}$	VGW Wave Heights (Ft)
$C_{D\ SKIRT}$	Skirt Drag Coefficient
X_{iHk}	Current inertial component of Hull Point along X_i (Ft)
Y_{iHk}	Current inertial component of Hull Point along Y_i (Ft)
Z_{iHk}	Current inertial component of Hull Point along Z_i (Ft)
ΔX_{i_n}	Vehicle Change in inertial position of each history state along X_i (Ft)
ΔY_{i_n}	Vehicle Change in inertial position of each history state along Y_i (Ft)
ΔZ_{i_n}	Vehicle Change in inertial position of each history state along Z_i (Ft)
ΔX_{iH_n}	Change in inertial position of hull pt. along X_i (Ft)
ΔY_{iH_n}	Change in inertial position of hull pt. along Y_i (Ft)
ΔZ_{iH_n}	Change in inertial position of hull pt. along Z_i (Ft)
$C^{i}_{v_n}$	Inertial to vehicle transformation for each history state
ΔX_{vH_n}	Change in position of hull pt. along X_b (Ft)
ΔY_{vH_n}	Change in position of hull point along Y_b (Ft)
ΔZ_{vH_n}	Change in position of hull point along Z_b (Ft)
R_{xn}	Magnitude of change of position of hull point referenced to CG along X_b (Ft)
R_{yn}	Magnitude of change of position of hull point referenced to CG along Y_b (Ft)
R_{zn}	Magnitude of change of position of hull point referenced to CG along Z_b (Ft)

X_{CG}	Positicn of CG measurcd along X_b = 30.0 Ft.
Y_{CG}	Position of CG measured along Y_b =-18.0 Ft.
Z_{CG}	Position of CG measured along Z_b = + 8.0 Ft.
K_{xy_n}	Kernel value for history states
N	State or kernel at t-NΔt
P_o	Nominal Pressure in Cushion Compartment =109.375 Lbs/Ft^2
ρ_w	Density of Water = 1.98 slugs/Ft^3
g	Gravitational acceleration = 32.17 Ft/Sec^2
a	Vehicle half length = 40 Ft.

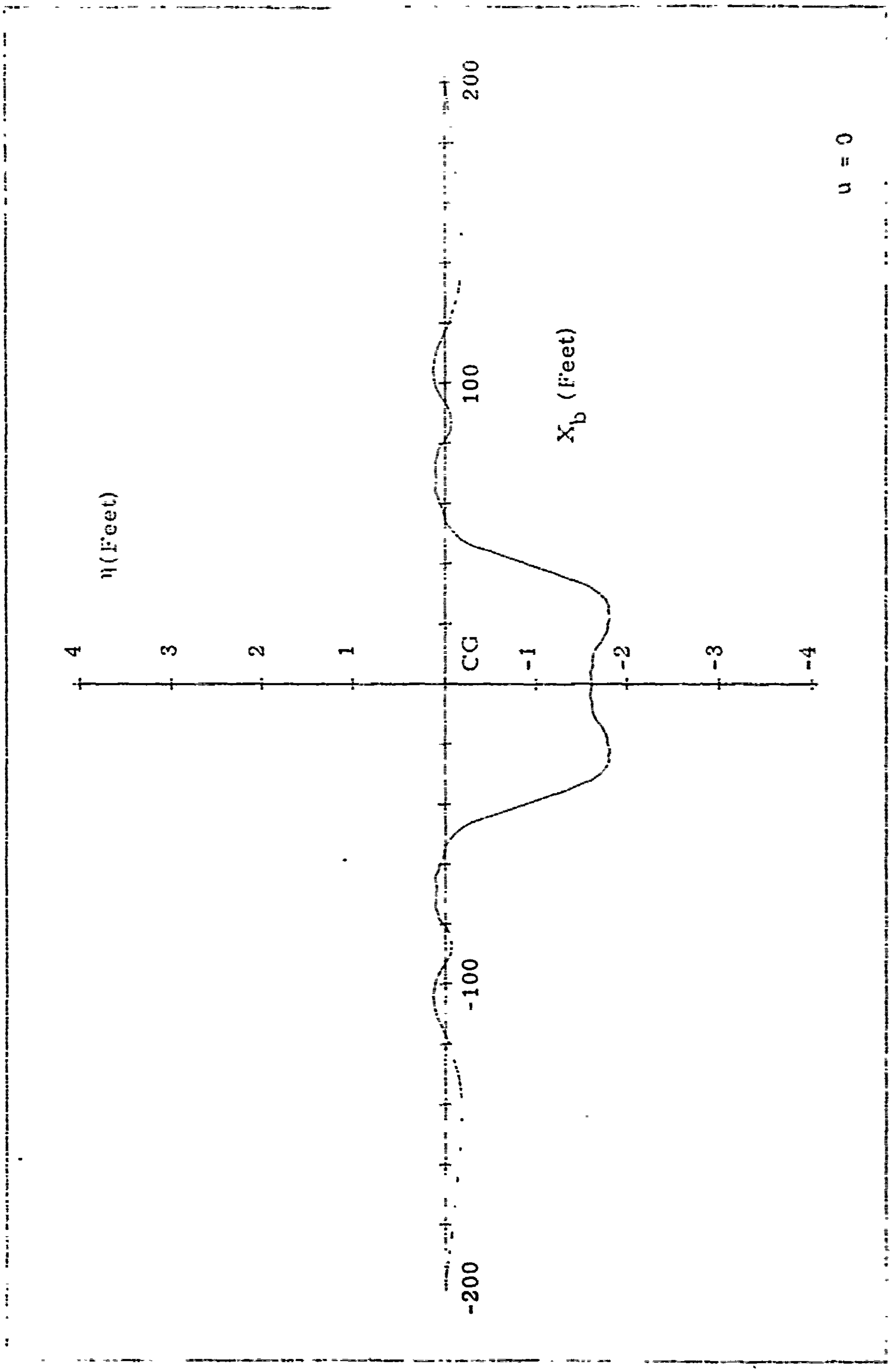


Figure 20. VGW Profile - u = 0, Longitudinal

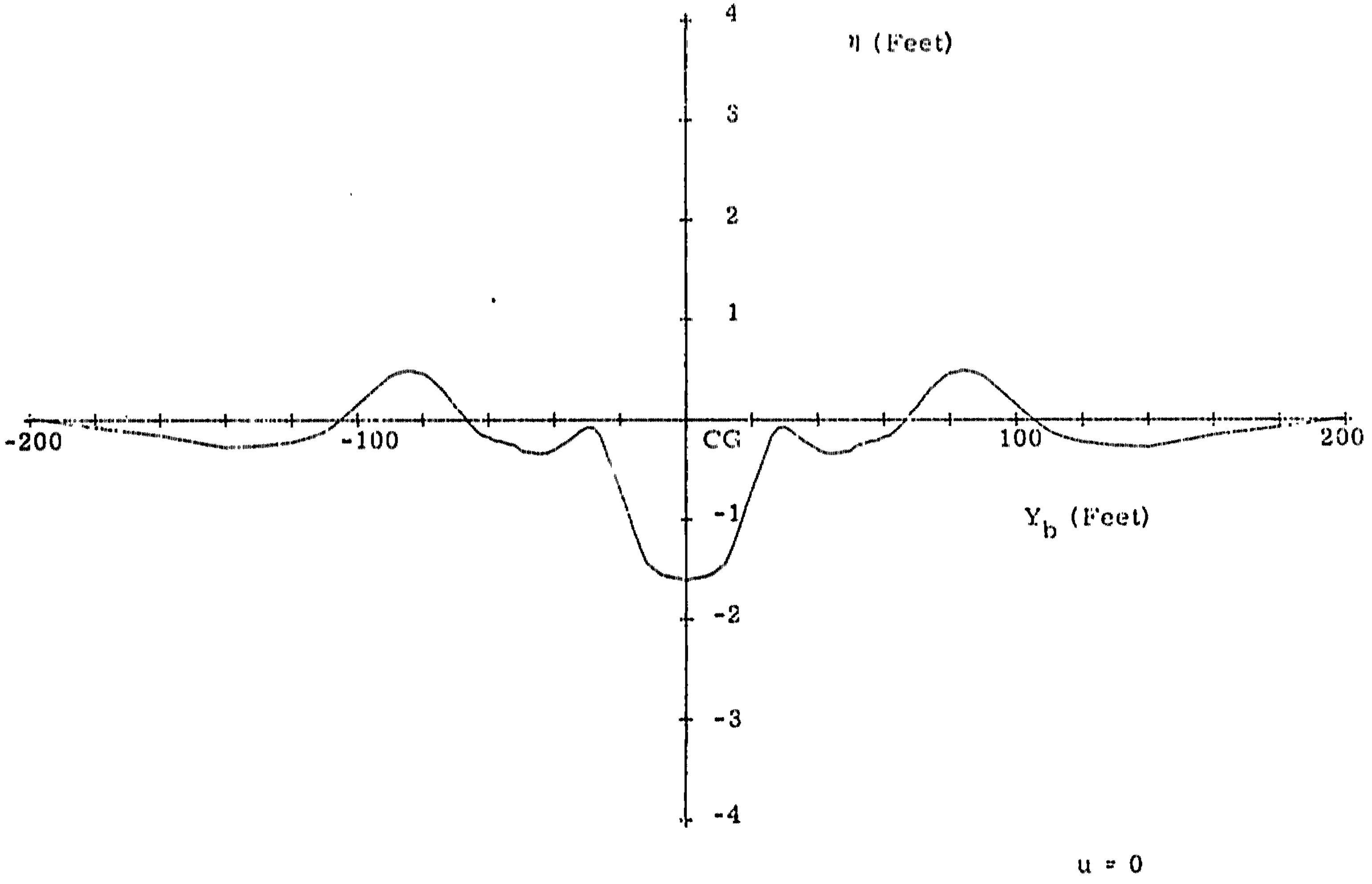


Figure 21. VGW Profile - u = 0, Lateral

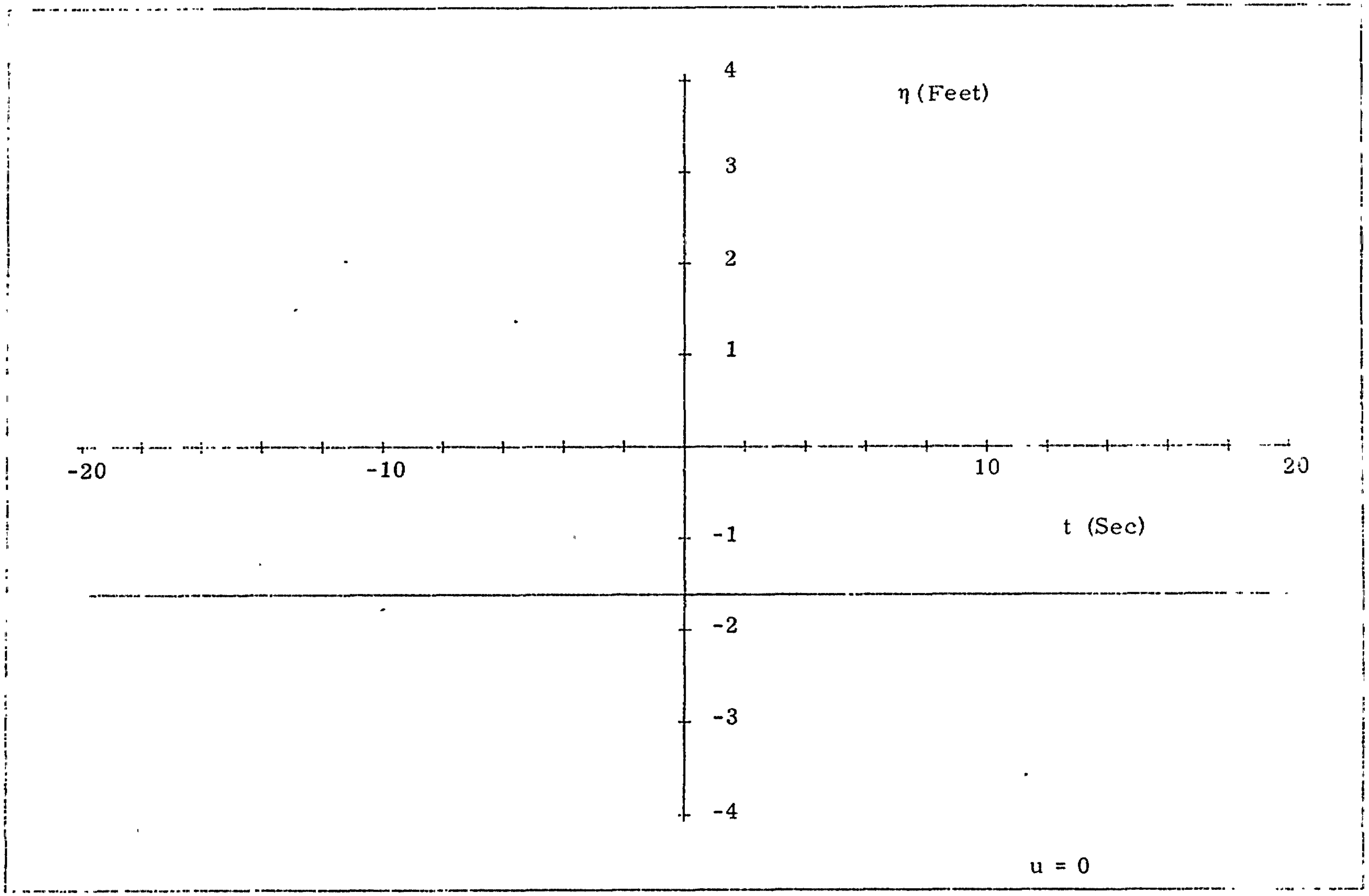


Figure 22. VGW Profile - u = 0, Time

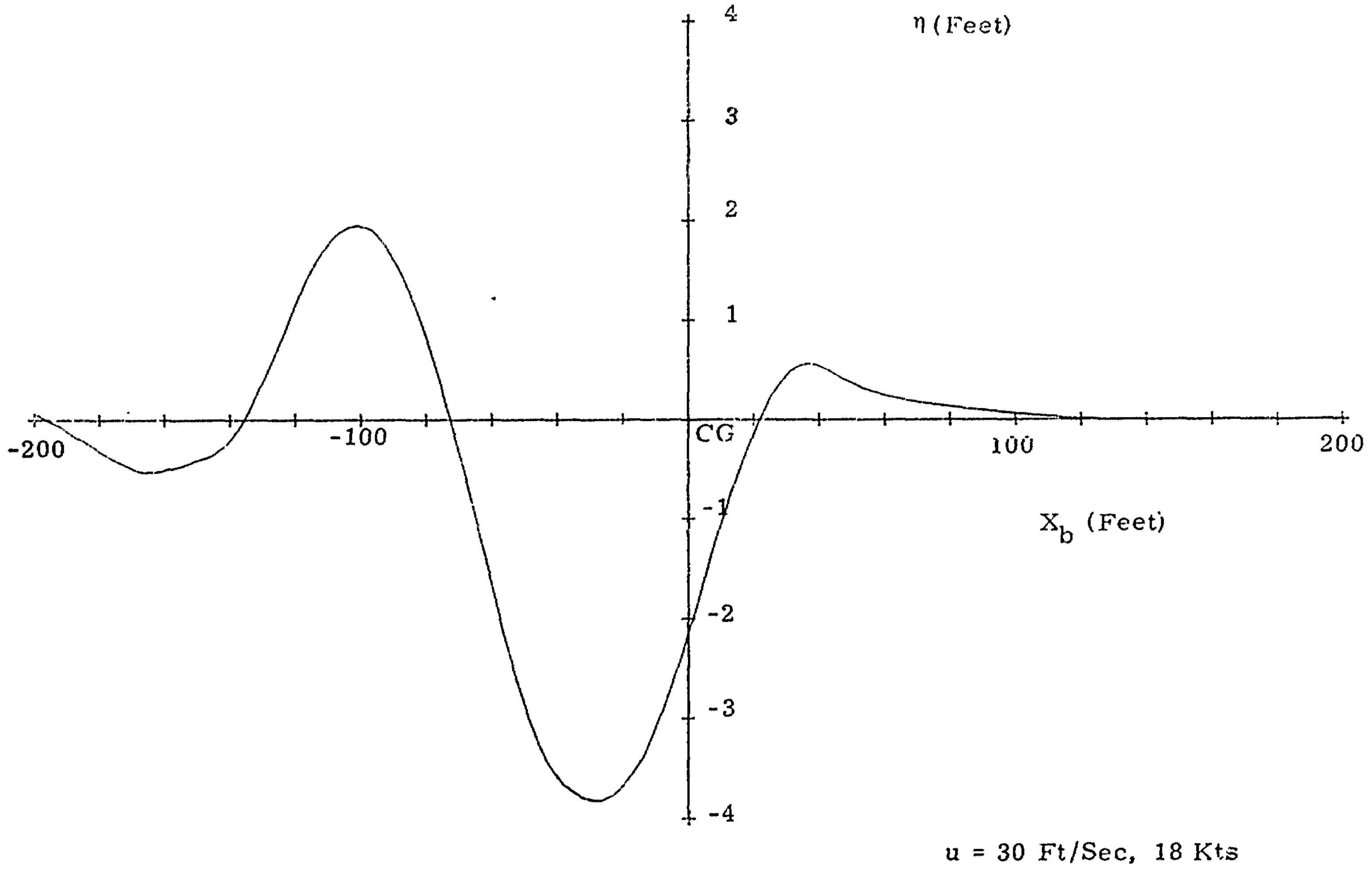


Figure 23. VGW Profile - u = 30 ft/sec, Longitudinal

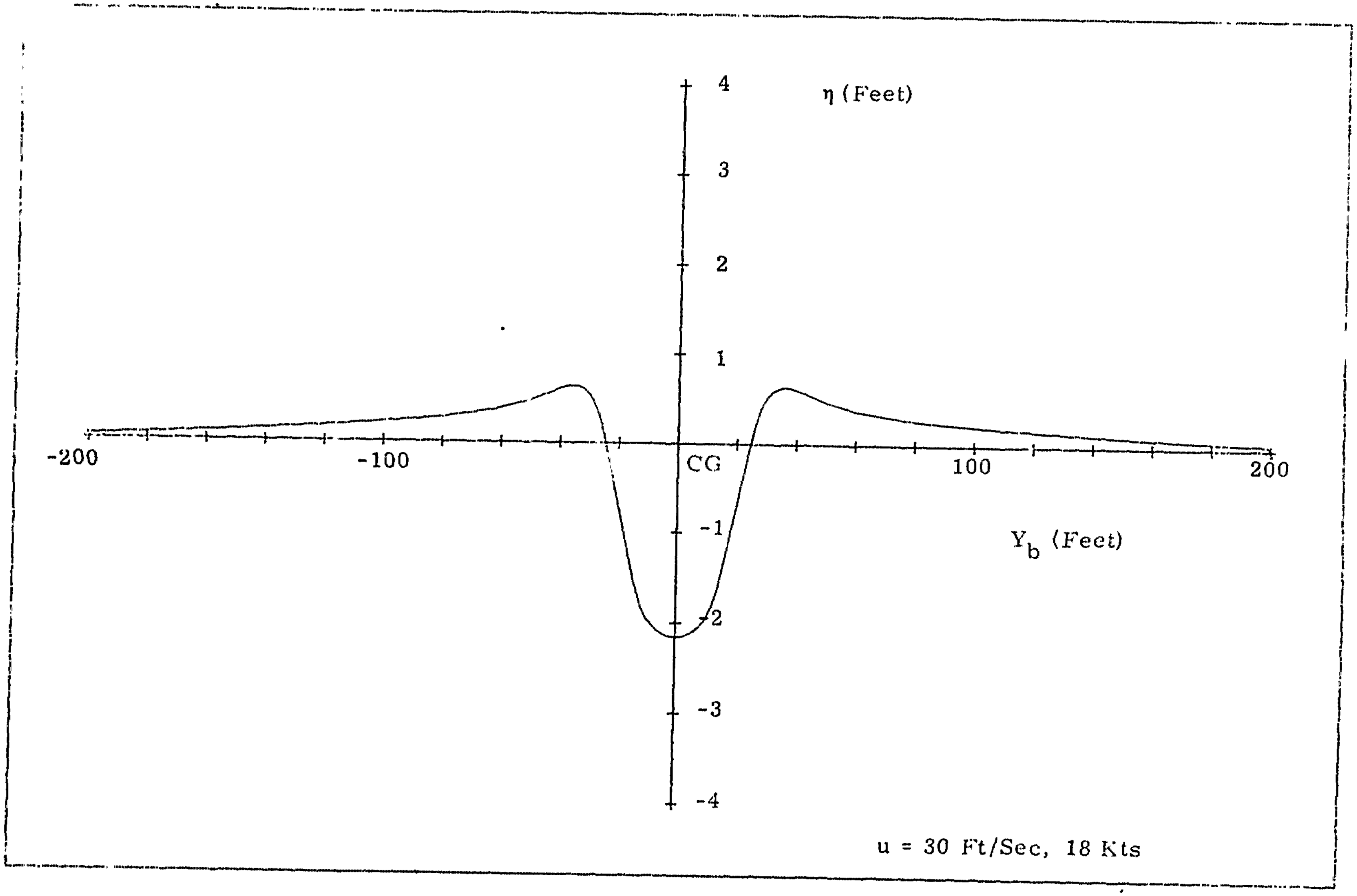


Figure 24. VGW Profile - u = 30 ft/sec, Lateral

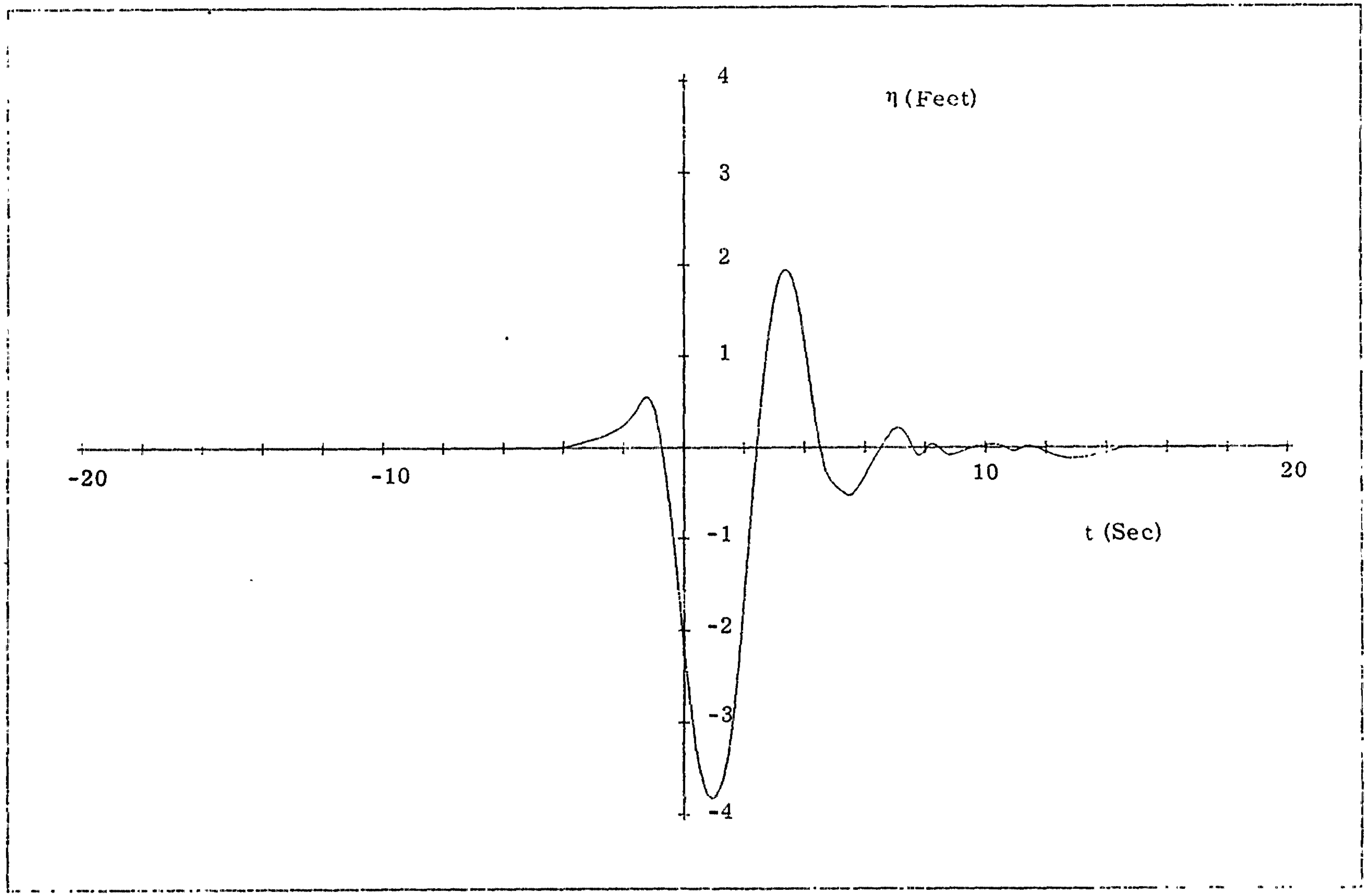


Figure 25. VGW Profile - u = 30 ft/sec, Time

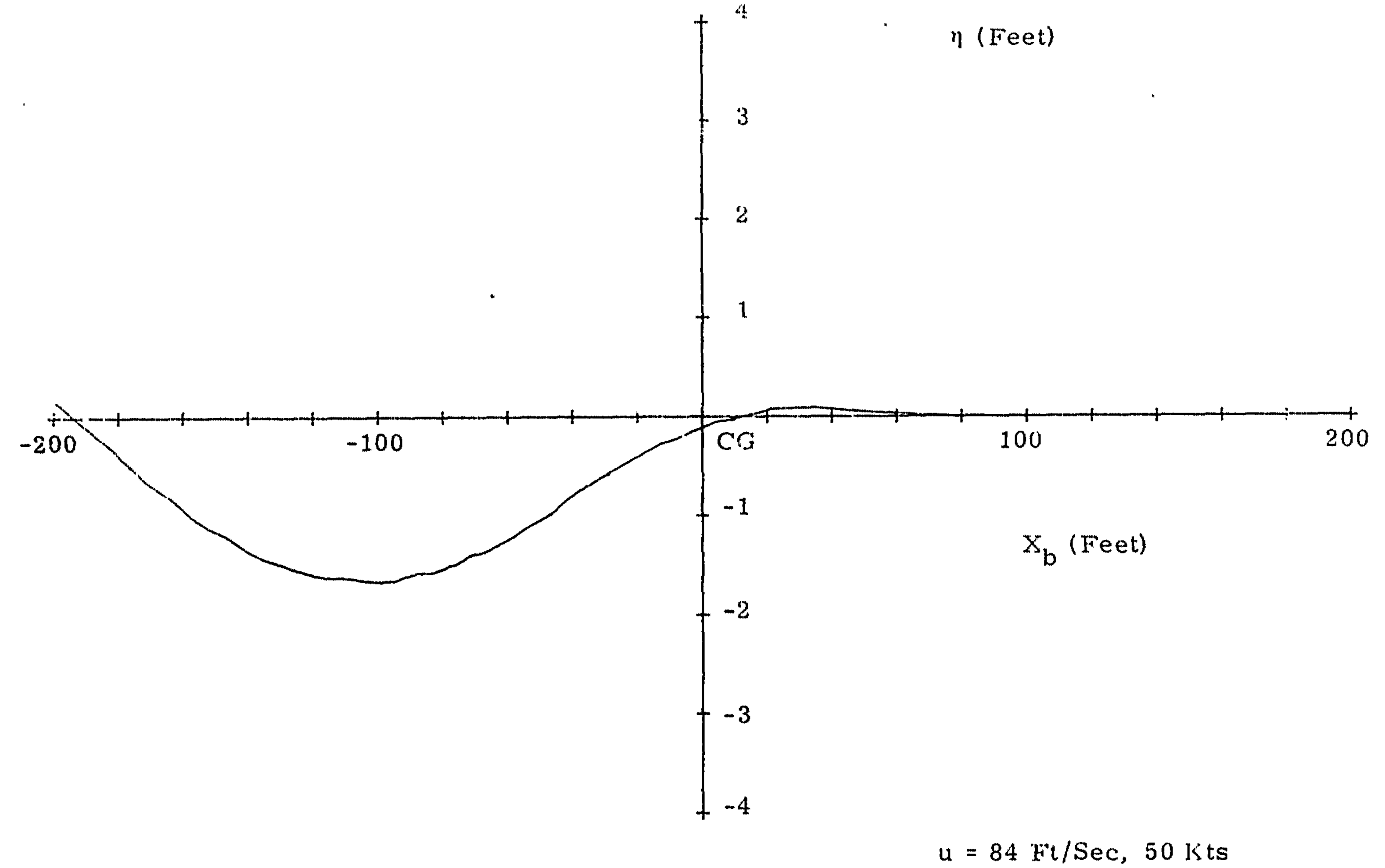
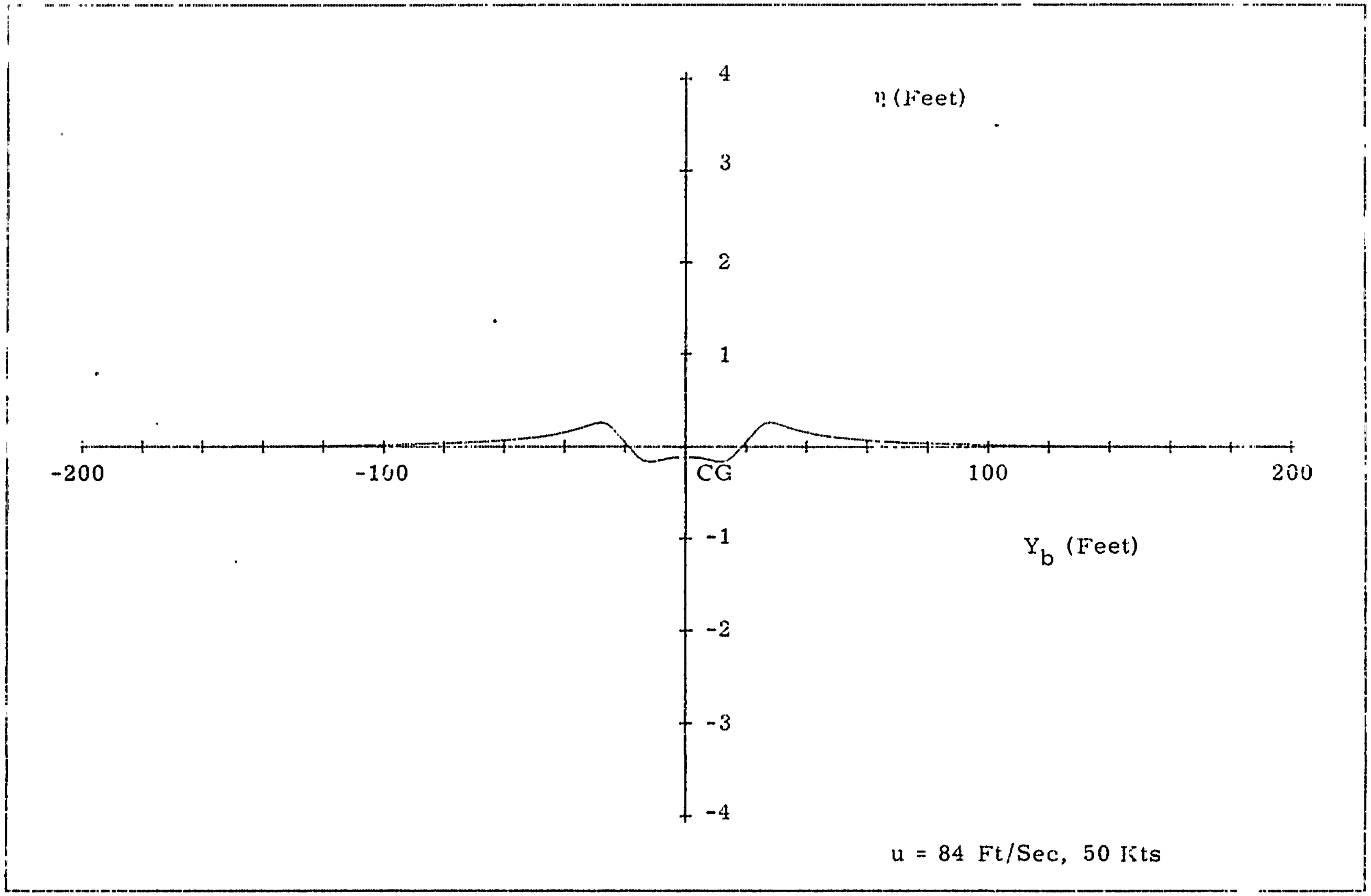
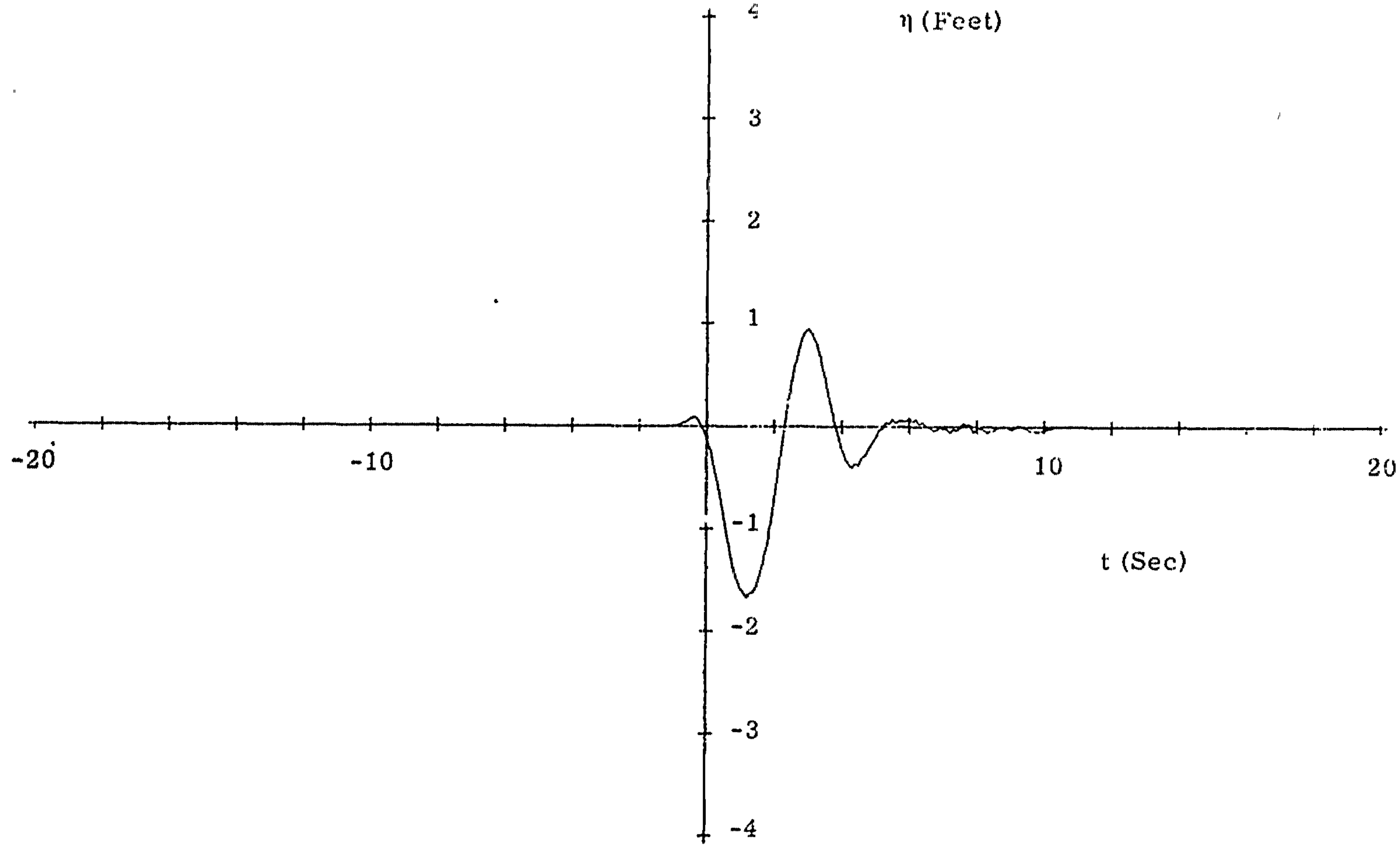


Figure 26. VGW Profile - u = 84 ft/sec. Longitudinal



u = 84 Ft/Sec, 50 Kts

Figure 27. VGW Profile - u = 84 ft/sec, Lateral



u = 84 Ft/Sec, 50 Kts

Figure 28. VGW Profile - u = 84 ft/sec, Time

KERNEL VALUE. The vehicle generated wave is created in the present and has existed for all time. The kernel function represents the nondimensionalized wave height at a particular instant in time due to an impulse in pressure. Therefore, the table consists of kernel values $K_{xy}(R_{xn}, R_{yn}, t-n\Delta t)$

versus X, Y position for a discrete time $(t-n\Delta t)$. With n known, it is held constant for the kernel value $K_{xy}(R_{xn}, R_{yn})$ determination. From the position of where the wave height is to be computed, the kernel value can be obtained.

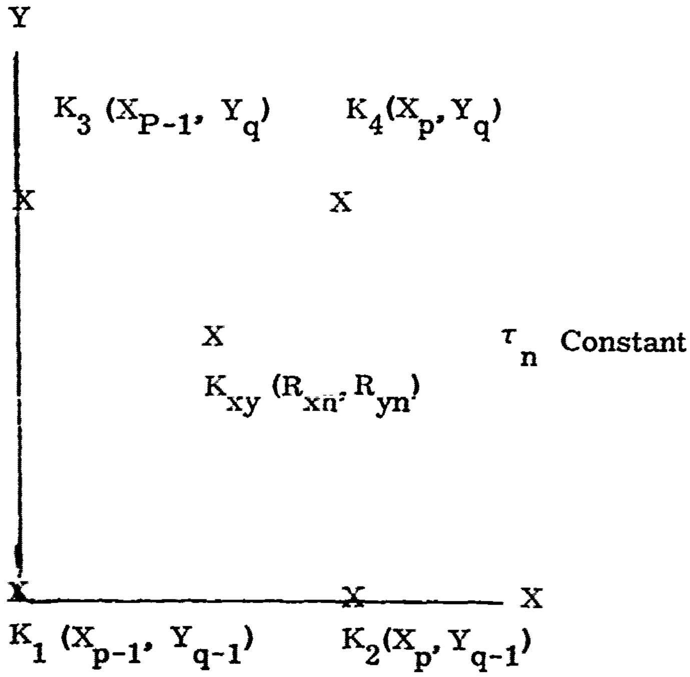
In the table, a particular kernel exists for discrete X, Y positions. The kernel function is assumed to be linear between tabulated points of known X, Y position. The particular R_x, R_y position may lie between other X, Y positions in the table. Linear interpolation is performed on the 4 kernel values at the 4 X, Y positions immediately surrounding the desired R_x, R_y.

Computer plots for the kernel function were generated for each unit of time from the VGW table. A family of 6 curves was plotted for each subtable of kernel (K_{xy}) versus longitudinal position (R_x) and lateral position (R_y). These plots were created by choosing alternate R_y's (R_x's) and holding the value of R_y (R_x) constant and then sweeping through each R_x (R_y) in the subtable. Nondimensionalized kernels were plotted versus position for times 2, 4, 6, 8 seconds corresponding to subtables numbered 22, 42, 62, 82 (see Figures 29, 30, 31, 32).

Subroutine: 1KINTRP

Retrieve K_{xy_n} for the function $K_{xy}\,(R_{xn}, R_{yn}, \tau_n)$

For each τ_n, values of the kernel function are pretabulated for each X, Y position.



Linear interpolation on K_1, K_2, K_3, K_4 to obtain K_{xy_n}

Weighting factors:

$$WT_{x1} = \frac{X_p - R_x}{X_p - X_{p-1}} \qquad WT_{x2} = 1 - WT_{x1}$$

$$WT_{y1} = \frac{Y_q - R_y}{Y_q - Y_{q-1}} \qquad WT_{y2} = 1 - WT_{y1}$$

$$K_{xy_n} = WT_{x1}\,WT_{y1}\,K_1 + WT_{x2}\,WT_{y1}\,K_2 + WT_{x1}WT_{y2}K_3 + WT_{x2}WT_{y2}K_4$$

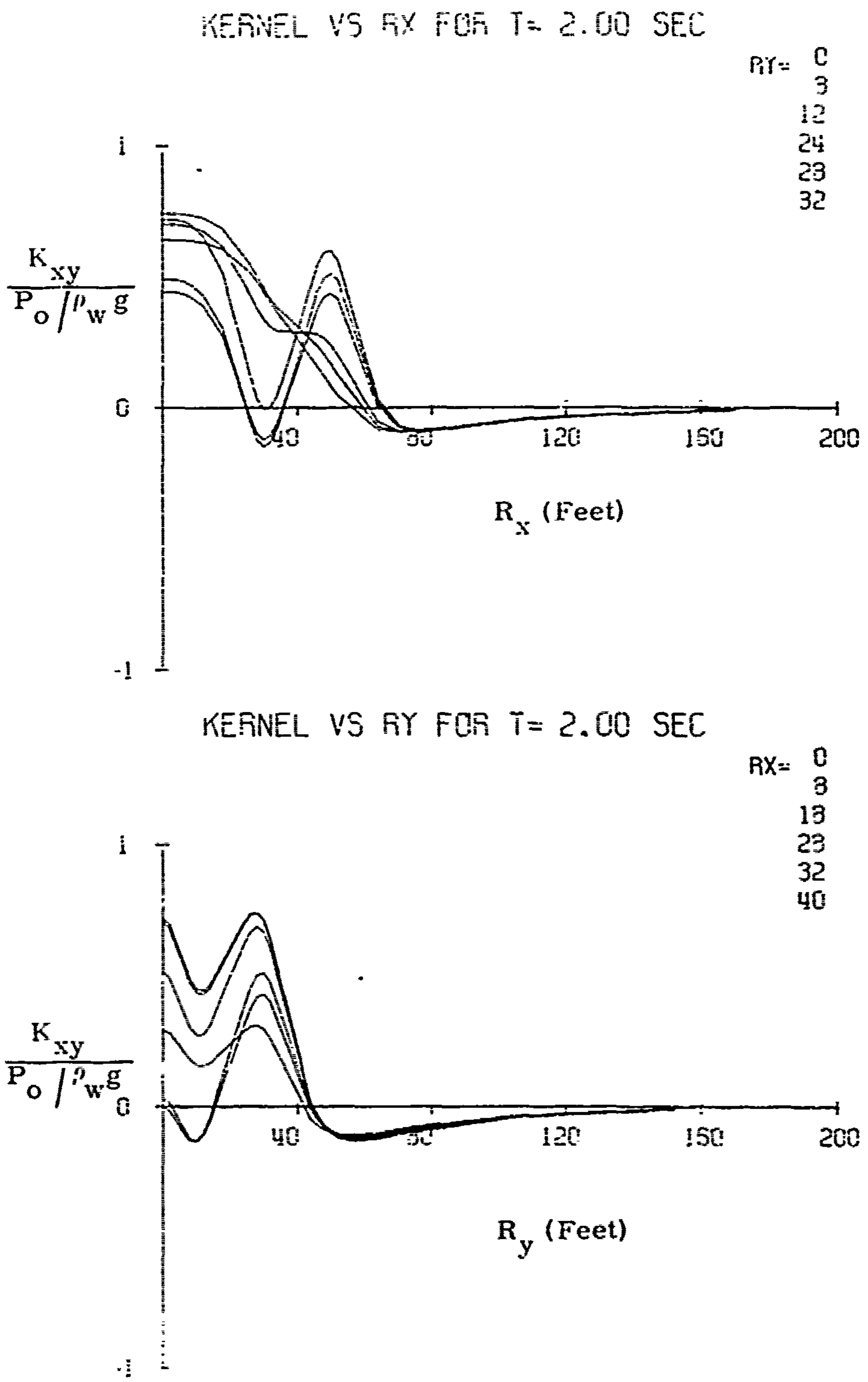


Table 22

Figure 29. VGW Kernel Profile at t = 2 sec

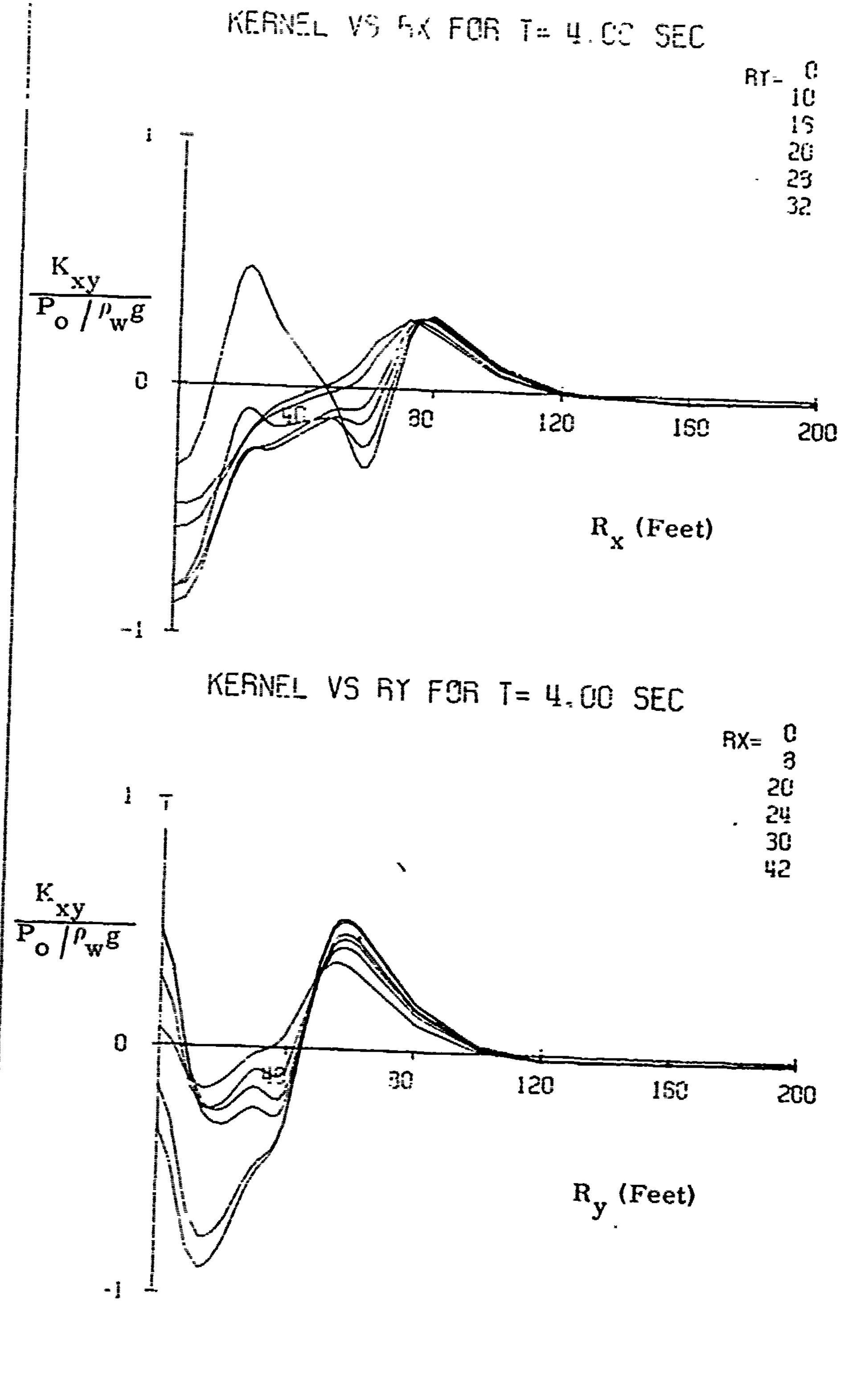


Table 42

Figure 30. VGW Kernel Profile at t = 4 sec

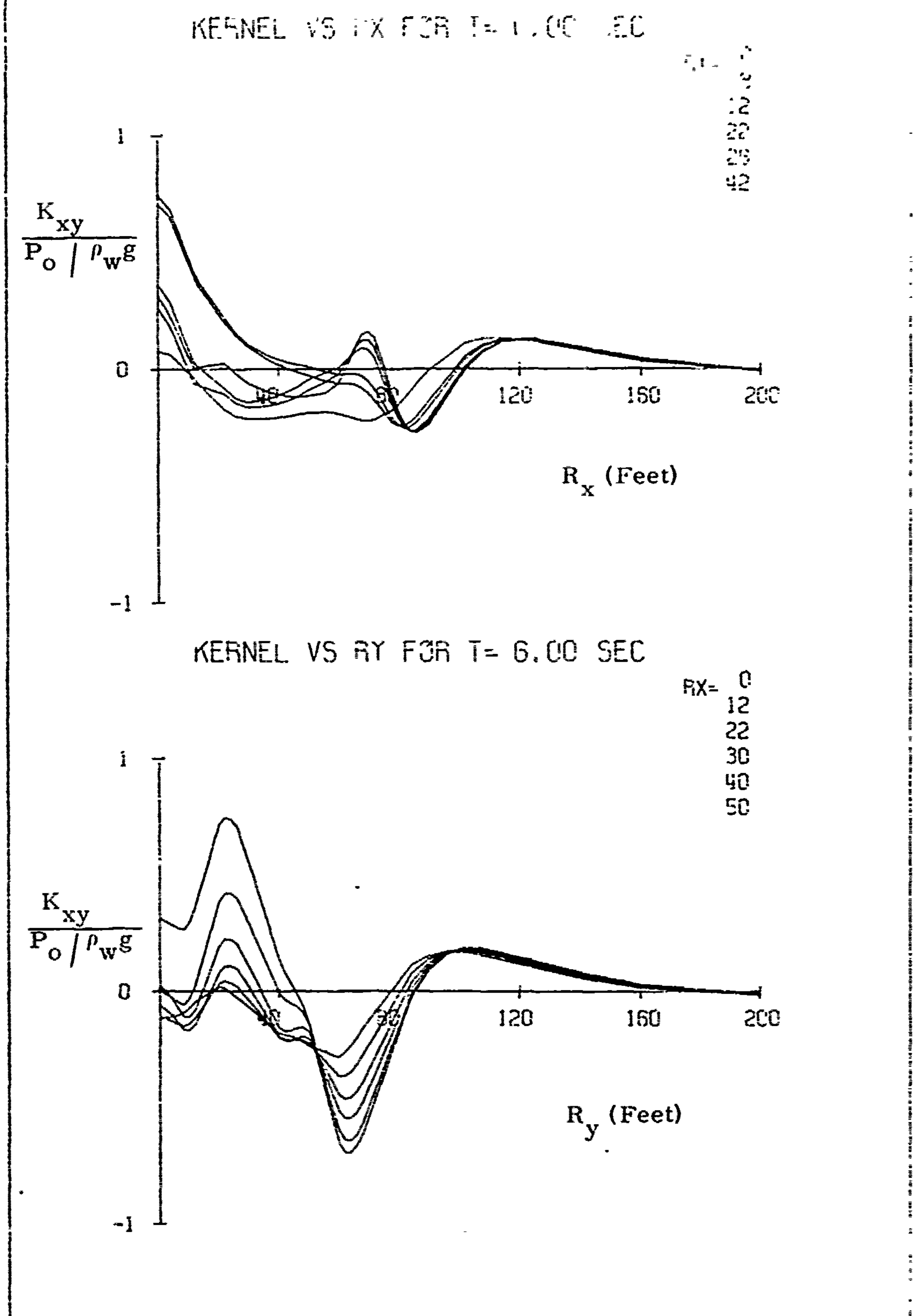


Table 62

Figure 31. VGW Kernel Profile at t = 6 sec

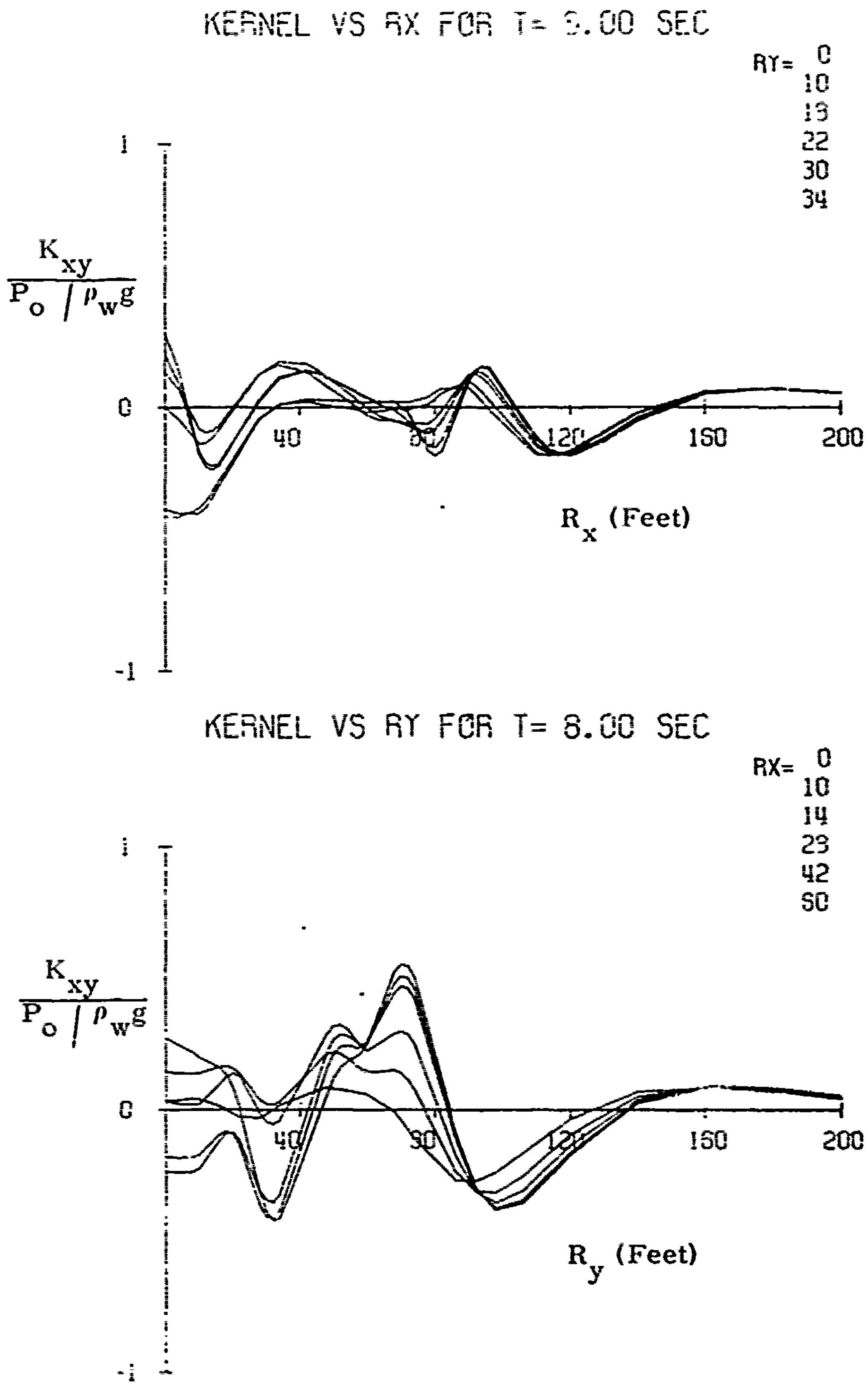


Table 82

Figure 32. VGW Kernel Profile at t = 8 sec

POSITION HISTORY OF ACV-LC. As has already been discussed, it is necessary to know all past positions of the vehicle in order to determine the vehicle generated wave (VGW) heights. However, it is not actually necessary to know the exact position, but the change in position for each time step up to the present. This position has 2 horizontal components, north and east. The variables are stored in a table:

ΔX_i change in north position for each time step

ΔY_i change in east position for each time step

The VGW model needs the body axis change in position, necessitating the saving of angles. With roll and pitch angles small, assume a change only in the horizontal plane and only save vehicle heading. Transformations need the direction cosines of the angles, however. Therefore, the variables are:

$C\psi$ Cos ψ at each time step

$S\psi$ Sin ψ at each time step

The table of past positions becomes successive entries of:

ΔX_i
ΔY_i
$C\psi$
$S\psi$

These positions are saved for $t-n\Delta t$, $n= 0, 1, 2, 3, \ldots, N_\tau$ where t is the present time step.

The position history over all time would consist of a voluminous amount of data that would exceed the core storage of the computer; as well as, the convolution over all time would be an impossible number of computations. The number of past position states has been limited to $N_\tau = 160$.

The position entries to the table are performed top down with the present position at the top. The true heading is inserted at each time step, but the linear change in position at the present time step has to be added to each previous change in position to yield the total change in position from the present to that particular time. For example, the table looks like the following for only a change in north position:

ΔX_{i_t}	0
$\Delta X_{i_{t-\Delta t}}$	ΔX_1
$\Delta X_{i_{t-2\Delta t}}$	ΔX_2

$\Delta X_{i_{t-3\Delta t}}$	ΔX_3
$\cdot$	$\cdot$
$\cdot$	$\cdot$
$\cdot$	$\cdot$
$\Delta X_{i_{t-(N_\tau -1)\Delta t}}$	$X_{N_\tau -1}$
$\Delta X_{i_{t-N_\tau \Delta t}}$	X_{N_τ}

At the next time step, $t+\Delta t$, $\Delta X_{i_{t+\Delta t}} = \Delta X_n$ is summed with all entries and the previous table becomes:

$$\begin{array}{l} 0 \\ \Delta X_n \\ \Delta X_1 + \Delta X_n \\ \Delta X_2 + \Delta X_n \\ \Delta X_3 + \Delta X_n \\ \cdot \\ \cdot \\ \cdot \\ \Delta X_{N_\tau -1} + \Delta X_n \end{array}$$

Where $\Delta X_{N_\tau} = \Delta X_{i_{t-N_\tau \Delta t}}$ has become $\Delta X_{N_\tau -1} + \Delta X_n$. The previous $\Delta X_{N\tau}$ is dropped from the bottom of the table and lost. This table has N_τ+1 entries, N_τ past positions plus the present position.

However, in the interest of limiting computations, the convolution integral need not be performed over all position history, but for selected time intervals. These positions are stored in an array (STATE) for $t-n\Delta t$: $n=0, 1, 2, 3, 4, 6, 8 \ldots, N_\tau$ The array length consists of $N_\tau /2 +2$ entries. The remaining position histories are stored in a separate entry (ODDSTATE) for $t-n\Delta t$:$n=5, 7, 9 \ldots, N_\tau -1$. The array length consists of $N_\tau /2 -2$ entries. This table storage and update is performed by Subroutine: HISTORY.

Initialization

The position table histories must be initialized in order to begin execution. All N_τ+1 entries are set equal to the initialized values of change in linear position and heading. These 4 quantities are stored successively throughout both tables:

$\Delta X_i = 0$

$\Delta Y_i = 0$

$C\psi$

$S\psi$

This initialization is performed in Subroutine: HSTRYIC.

TABLE GENERATION. The Vehicle Generated Wave (VGW) table consists of 82 subtables of distinct kernel values. Each subtable corresponds to a pressure wave versus x, y position for a discrete time in the past. The last subtable (number 82) corresponds to 8 seconds in the past. The x, y positions are referenced to the CG in body coordinates and are magnitude only.

The kernel table is generated by a FORTRAN program TABKS. The spline curve data points are read and a set of spline curve coefficients are computed. At which point, the program reads the data to begin table generation. This data consists of nondimensionalized time and a list of nondimensionalized x and y positions. For each x position, the y position list is swept to compute a kernel value for each combination of x, y position. The number of kernel values computed equals the number of x values times the number of y values. Consequently, the computational ability has been limited to 30 x's by 30 y's. After the subtable is computed, it is formatted and written on magnetic tape. Data for time and position is then read for the next subtable computation and the calculations are repeated for a new time.

These subtables were checked for curve smoothness by an eyeball method. Two computer plots were generated per subtable, one versus R_x and the other versus R_y. The fifth R_x (R_y) in the subtable was randomly chosen as representative and all R_y's (R_x's) were swept. These plots were visually selected on piecewise linear curve fit. Discontinuities were reduced by selectively adding an R_x (R_y) to the input position data. The subtable was recreated by only calculating the kernel values which did not already appear in the subtable. The final check of curve smoothness was accomplished by plotting a family of 6 curves per subtable (see Section 3.5.2) and visually determining the necessity of adding other points.

TABKS: Compute kernel values from the equation

$$K^S (R_x, R_y, t) = \int_0^\infty \frac{k\,dk}{\alpha^2} \gamma \sin \gamma t \left[\sum_{i=1}^{2} \sum_{j=1}^{2} (-1)^{i+j} Q(k, R_{ij}, \psi_{ij}) \right]$$

t is non-dimensionalized time

where R_x and R_y are the non-dimension coordinates of the point where the kernel is being computed relative to the vehicle cg. in the vehicle coordinate system at non-dimensional time t

k is the radial component of the conversion of the two dimensional inverse fourier transform to polar coordinates

α is a diffusion parameter (set to 0.5)

$\gamma = \sqrt{g\ k\ \tanh(kh)}$

g = acceleration of gravity

h = non-dimensional depth of water (assumed 1.0)

R_{ij} and ψ_{ij} are polar coordinates of (R_x, R_y) referred to the four corners of the craft

$\sum_{i=1}^{2} \sum_{j=1}^{2} (-1)^{i+j} Q(k, R_{ij}, \psi_{ij})$ integral of the angular component of the inverse Fourier transform

FINDK0

Determine a limit of integration (k_o) to replace the infinite limit, such that

$$\epsilon \le 2 \left(\frac{2}{\pi}\right)^2 \sqrt{k_o}\ e^{-\frac{\pi}{2\alpha} k_o}$$

where ϵ is the maximum allowable error (set to .01)

INTDK2

Perform the integration over k using a Gauss-quadrature method.

WNING

Compute the integrand

$$k \gamma \sin(\gamma t) \left[\sum_{i=1}^{2} \sum_{j=1}^{2} (-1)^{i+j} Q(k, R_{ij}, \psi_{ij}) \right]$$

Q24

Perform the double sum and compute the function

$$Q(k, R_{ij}, \psi_{ij}) = -\sum_{n=1,3,5,\ldots}^{\infty} C_n(k') J_{2n}(kR_{ij}) \sin(2n\psi_{ij})$$

where $k' = \alpha' k$

and $\alpha' = \pi/2\alpha$

J_{2n} = $2n^{th}$ order Bessel function

$C_n(k')$ are coefficients which are fitted with a spline curve

PREP: Read in coefficients of spline curves.

GCNK4: Compute $C_n(k')$ from spline coefficients

BESSEL: Compute the Bessel function $J_{2n}(kR)$ in double precision

SESSEL: Compute the Bessel function $J_{2n}(kR)$ in single precision

VGW CALCULATIONS. It has already been stated that there are a voluminous number of calculations to determine all VGW heights under the hull planform. So many that the calculations cannot be performed in real-time. The VGW height calculations are performed in the background and are processed through a first order lag in real-time to prevent a step in VGW height.

Subroutine: FILTER

$$\eta_{VGW_k} = \frac{1}{\tau_{VGW} S + 1} \quad \eta_{VGW_{NEW_k}}$$

$$\eta_{VGW_k} = \frac{\Delta t}{\tau_{VGW}} \quad \eta_{VGW_{NEW_K}} + \left[1 - \frac{\Delta t}{\tau_{VGW}}\right] \eta_{VGW_k}$$

k Number of hull bottom plating points

Δt Sampling period = .05 Sec

τ_{VGW} First order lag time constant = .05 Sec

η_{VGW_k} Vehicle generated wave height (Ft)

$\eta_{VGW_{NEW_k}}$ Newly calculated VGW height (Ft)

STATE INTEGRATIONS

An orthogonal Euler transformation matrix is defined (see table 8)to relate body coordinates of the vehicle to the earth-fixed coordinate system. With the transformations defined, the Basic Motion Equations (see table 9)can also be defined to show how all positions, rates, and accelerations are related.

Two integrations are necessary to take the output of the Equations of Motion (acceleration) to arrive at position. These are accomplished by a second order predictor-corrector numerical integration. This scheme executes very quickly and only needs a single execution of the Equations of Motion. The stability and accuracy of the predictor-corrector are very good for such short computation times.

The second order predictor-corrector method used is shown in one variable to facilitate ease in reading. The scheme in all variables is also shown for completeness.

The integration scheme simply predicts position and velocity, corrects position and velocity, corrects position, solves the Equations of Motion, and corrects velocity. The velocity integrations are performed in the earth-fixed coordinate system. Summing initial conditions with the integrated values is also necessary.

Because the scheme is of second order, two past values are needed for each integration. A simple Euler method is used to get the scheme started, along with an initialization pass.

TABLE 8. EULER ROTATION DEFINITION

Subroutine: C THREE

Coordinate Transformation- Inertial to Vehicle

$$C_{v}^{i} = \left[\phi\right]_{x}\left[\Theta\right]_{y}\left[\psi\right]_{z} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

$$= \begin{bmatrix} C\Theta C\psi & C\Theta S\psi & -S\Theta \\ -C\phi S\psi + S\phi S\Theta C\psi & C\phi C\psi + S\phi S\Theta S\psi & S\phi C\Theta \\ S\phi S\psi + C\phi S\Theta C\psi & -S\phi C\psi + C\phi S\Theta S\psi & C\phi C\Theta \end{bmatrix}$$

$$\approx \begin{bmatrix} C\psi & S\psi & -\Theta \\ -S\psi & C\psi & \phi \\ \phi S\psi + \Theta C\psi & -\phi C\psi + \Theta S\psi & 1 \end{bmatrix}$$

For Small ϕ, θ ($< 10^{o}$)

ϕ	Euler Angle Roll of ACV (RAD)
Θ	Euler Angle Pitch of ACV (RAD)
ψ	Euler Angle Heading of ACV (DEG)
$S\psi$	Sin ψ
$C\psi$	Cos ψ

TABLE 9 . BASIC MOTION EQUATIONS

Subroutine

$$\begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} = \begin{Bmatrix} {}_o\int^t \dot{u}\,dt \\ {}_o\int^t \dot{v}\,dt \\ {}_o\int^t \dot{w}\,dt \end{Bmatrix} \quad \text{INTGRT} \quad \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix} = \begin{Bmatrix} {}_o\int^t \dot{p}\,dt \\ {}_o\int^t \dot{q}\,dt \\ {}_o\int^t \dot{r}\,dt \end{Bmatrix},$$

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} + \begin{Bmatrix} u_o \\ v_o \\ w_o \end{Bmatrix} \quad \text{SUMICS} \quad \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix} + \begin{Bmatrix} p_o \\ q_o \\ r_o \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{Bmatrix} = C_i^v \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad \text{C THREE} \quad \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} \approx \begin{Bmatrix} p + \theta r \\ q - \phi r \\ \phi q + r \end{Bmatrix}$$

$$\begin{Bmatrix} X'_i \\ Y'_i \\ Z'_i \end{Bmatrix} = \begin{Bmatrix} {}_o\int^t \dot{X}_i\,dt \\ {}_o\int^t \dot{Y}_i\,dt \\ {}_o\int^t \dot{Z}_i\,dt \end{Bmatrix} \quad \text{INTGRT} \quad \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} = \begin{Bmatrix} {}_o\int^t \dot{\phi}\,dt \\ {}_o\int^t \dot{\theta}\,dt \\ {}_o\int^t \dot{\psi}\,dt \end{Bmatrix}$$

$$\begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{Bmatrix} X'_i \\ Y'_i \\ Z'_i \end{Bmatrix} + \begin{Bmatrix} X_{io} \\ Y_{io} \\ Z_{io} \end{Bmatrix} \quad \text{SUMICS} \quad \begin{Bmatrix} \Phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \Phi' \\ \theta' \\ \psi' \end{Bmatrix} + \begin{Bmatrix} \Phi_o \\ \theta_o \\ \psi_o \end{Bmatrix}$$

SECOND ORDER PREDICTOR - CORRECTOR INTEGRATIONS

Scheme in One Variable

For $\underline{t=0,}$ Where $X_t = U_t = 0$

$$X = X_o + X_t$$

$$U = U_o + U_t$$

$$\dot{X}_t = C_i^v \, U$$

$$X_{t-2\Delta t} = X_t$$

$$\dot{X}_{t-\Delta t} = \dot{X}_t$$

$$\Delta PC = 0$$

Gravity Forces and Moments

Sum Forces and Moments

Equations of Motion

$$U_{t-2\Delta t} = U_t$$

$$\dot{U}_{t-\Delta t} = 0$$

$$\Delta\dot{P}C = 0$$

For $\underline{t = \Delta t}$

$$X_t = \frac{\Delta t}{2} \dot{X}_{t - \Delta t}$$

$$U_t = \frac{\Delta t}{2} \dot{U}_{t - \Delta t}$$

$$X = X_o + X_t$$

$$U = U_o + X_t$$

$$\dot{X}_t = C_i^v U$$

$$X_t = \frac{4}{5} \left[\frac{\Delta t}{2} (\dot{X}_t + \dot{X}_{t - \Delta t}) + X_{t - 2\Delta t} \right]$$

$$X = X_o + X_t$$

Gravity Forces and Moments

Sum Forces and Moments

Equations of Motion

$$U_t = \frac{4}{5} \left[\frac{\Delta t}{2} (\dot{U}_t + \dot{U}_{t - \Delta t}) + U_{t - 2\Delta t} \right]$$

$$U = U_o + U_t$$

$$\dot{X}_t = C_i^v U$$

$$U_{t - \Delta t} = U_t$$

$$\dot{U}_{t - \Delta t} = \dot{U}_t$$

$$X_{t - \Delta t} = X_t$$

$$\dot{X}_{t - \Delta t} = \dot{X}_t$$

For $\underline{t = n\Delta t}$, $n > l$

$$P_t = X_{t-2\Delta t} + 2\Delta t\,\dot{X}_{t-\Delta t}$$

$$P_t = U_{t-2\Delta t} + 2\Delta t\,\dot{U}_{t-\Delta t}$$

$$X_t = P_t + \Delta PC$$

$$U_t = \dot{P}_t + \Delta\dot{P}C$$

$$X = X_o + X_t$$

$$U = U_o + U_t$$

$$\dot{X}_t = C_i^v u$$

$$C_t = X_{t-\Delta t} + \frac{\Delta t}{2}\left(\dot{X}_t + \dot{X}_{t-\Delta t}\right)$$

$$\Delta PC = 4/5\,(C_t - P_t)$$

$$X_t = P_t + \Delta PC$$

$$X = X_o + X_t$$

$$X_{t-2\Delta t} = X_{t-\Delta t}$$

$$X_{t-\Delta t} = X_t$$

Gravity Forces and Moments

Sum Forces and Moments

Equations of Motion

$$\dot{C}_t = U_{t-\Delta t} + \frac{\Delta t}{2} (\dot{U}_t + \dot{U}_{t-\Delta t})$$

$$\Delta\dot{P}C = 4/5 (\dot{C}_t - \dot{P}_t)$$

$$U_t = \dot{P}_t + \Delta\dot{P}C$$

$$U = U_o + U_t$$

$$\dot{X}_t = C_i^v \; U$$

$$U_{t-2\Delta t} = U_{t-\Delta t}$$

$$U_{t-\Delta t} = U_t$$

$$\dot{U}_{t-\Delta t} = \dot{U}_t$$

$$\dot{X}_{t-\Delta t} = \dot{X}_t$$

X_o	Initial Inertial Position
U_o	Initial Body Axis Velocity
X_t	Integrated Inertial Velocity
U_t	Integrated Body Axis Acceleration
X	Inertial Position
U	Body Axis Velocity
C_i^v	Vehicle to Inertial Transformation Matrix
$\dot{X}_t$	Inertial Velocity
$X_{t-2\Delta t}$	Inertial Position at 2 Time Periods Past
$\dot{X}_{t-\Delta t}$	Inertial Velocity at 1 Time Period Past
ΔPC	Position Corrector Value
$U_{t-2\Delta t}$	Body Axis Velocity at 2 Time Periods Past
$\Delta \dot{P}C$	Velocity Corrector Value
Δt	Sampling Period
$\dot{U}_t$	Body Axis Acceleration
$U_{t-\Delta t}$	Body Axis Velocity at 1 Time Period Past
$X_{t-\Delta t}$	Inertial Position at 1 Time Period Past
P_t	Predicted Inertial Position
$\dot{P}_t$	Predicted Body Axis Velocity
C_t	Corrected Inertial Position
$\dot{C}_t$	Corrected Body Axis Velocity
$\dot{U}_{t-\Delta t}$	Body axis acceleration at 1 time period past

Change in Inertial Position for Vehicle Generated Wave

For $\underline{t = 0}$

$$\Delta X_i = 0$$

$$\Delta Y_i = 0$$

For $\underline{t = \Delta t}$

$$\Delta X_i = \Delta X_{i_{t-\Delta t}} = X_t$$

$$\Delta Y_i = \Delta Y_{i_{t-\Delta t}} = Y_t$$

For $\underline{t = n\Delta t}$, $n > 1$

$$\Delta X_i = \frac{1}{5}\left[2\Delta t\,(2\,\dot{X}_{t-\Delta t} + \dot{X}_i) - \Delta X_{i_{t-\Delta t}}\right]$$

$$\Delta Y_i = \frac{1}{5}\left[2\,\Delta t\,(2\,\dot{Y}_{t-\Delta t} + \dot{Y}_i) - \Delta Y_{i_{t-\Delta t}}\right]$$

$$\Delta X_{i_{t-\Delta t}} = \Delta X_i$$

$$\Delta Y_{i_{t-\Delta t}} = \Delta Y_i$$

ΔX_i	Change in position along X_i (Ft)
ΔY_i	Change in position along Y_i (Ft)
$\Delta X_{i_{t-\Delta t}}$	ΔX_i at one sampling period past (Ft)
$\Delta Y_{i_{t-\Delta t}}$	ΔY_i at one sampling period past (Ft)

Scheme in Total

For <u>t = 0,</u> Where

Subroutine

$$\begin{Bmatrix} X'_i \\ Y'_i \\ Z'_i \end{Bmatrix} = \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} = \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} = \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix} = 0$$

SYSACVIC

Subroutine: INTGRT

$$\begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{Bmatrix} X_{io} \\ Y_{io} \\ Z_{io} \end{Bmatrix} + \begin{Bmatrix} X'_i \\ Y'_i \\ Z'_i \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \phi_o \\ \theta_o \\ \psi_o \end{Bmatrix} + \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} \qquad -180^{o} \leq \psi \leq 180$$

SUMICS

$$\begin{Bmatrix} \phi_R \\ \theta_R \\ \psi_R \end{Bmatrix} = \frac{\pi}{180} \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u_o \\ v_o \\ w_o \end{Bmatrix} + \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix}$$

SUMICS

Subroutine

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} p_o \\ q_o \\ r_o \end{Bmatrix} + \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{Bmatrix} = C_i^v \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

CTHREE

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{Bmatrix} p + \theta r \\ q - \phi r \\ \phi q + r \end{Bmatrix}$$

CTHREE

$$\begin{Bmatrix} X_{t-2\Delta t} \\ Y_{t-2\Delta t} \\ Z_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix}$$

$$\begin{Bmatrix} \phi_{t-2\Delta t} \\ \theta_{t-2\Delta t} \\ \psi_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{X}_{t-\Delta t} \\ \dot{Y}_{t-\Delta t} \\ \dot{Z}_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{Bmatrix}$$

Subroutine

$$\begin{Bmatrix} \dot{\phi}_{t-\Delta t} \\ \dot{\theta}_{t-\Delta t} \\ \dot{\psi}_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta PC_x \\ \Delta PC_y \\ \Delta PC_z \end{Bmatrix} = 0$$

$$\begin{Bmatrix} \Delta PC_\phi \\ \Delta PC_\theta \\ \Delta PC_\psi \end{Bmatrix} = 0$$

Gravity Forces and Moments GRAVITY

Sum Forces and Moments FORCES

Equations of Motion MOTION

$$\begin{Bmatrix} u_{t-2\Delta t} \\ v_{t-2\Delta t} \\ w_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix}$$

$$\begin{Bmatrix} p_{t-2\Delta t} \\ q_{t-2\Delta t} \\ r_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix}$$

$$\left\{\begin{matrix}\dot{u}_{t-\Delta t}\\ \dot{v}_{t-\Delta t}\\ \dot{w}_{t-\Delta t}\end{matrix}\right\} = 0$$

$$\left\{\begin{matrix}\dot{p}_{t-\Delta t}\\ \dot{q}_{t-\Delta t}\\ \dot{r}_{t-\Delta t}\end{matrix}\right\} = 0$$

$$\left\{\begin{matrix}\Delta PC_{u}\\ \Delta PC_{v}\\ \Delta PC_{w}\end{matrix}\right\} = 0$$

$$\left\{\begin{matrix}\Delta PC_{p}\\ \Delta PC_{q}\\ \Delta PC_{r}\end{matrix}\right\} = 0$$

For t=Δt

Subroutine

$$\begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix} = \frac{\Delta t}{2} \begin{Bmatrix} \dot{X}_{t-\Delta t} \\ \dot{Y}_{t-\Delta t} \\ \dot{Z}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} = \frac{\Delta t}{2} \begin{Bmatrix} \dot{\phi}_{t-\Delta t} \\ \dot{\theta}_{t-\Delta t} \\ \dot{\psi}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{Bmatrix} X_{io} \\ Y_{io} \\ Z_{io} \end{Bmatrix} + \begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \phi_o \\ \theta_o \\ \psi_o \end{Bmatrix} + \begin{Bmatrix} \phi' \\ \theta' \\ \dot{\psi}' \end{Bmatrix} \quad -180^\circ \leq \psi \leq 180^\circ$$

SUMICS

$$\begin{Bmatrix} \phi_R \\ \theta_R \\ \psi_R \end{Bmatrix} = \frac{\pi}{180} \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}$$

SUMICS

Subroutine

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u_o \\ v_o \\ w_o \end{Bmatrix} + \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} p_o \\ q_o \\ r_o \end{Bmatrix} + \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{Bmatrix} = C_i^v \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

CTHREE

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{Bmatrix} p + \theta r \\ q - \phi r \\ \phi q + r \end{Bmatrix}$$

CTHREE

$$\begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix} = \frac{4}{5} \left[\frac{\Delta t}{2} \begin{Bmatrix} \dot{X}_i + \dot{X}_{t-\Delta t} \\ \dot{Y}_i + \dot{Y}_{t-\Delta t} \\ \dot{Z}_i + \dot{Z}_{t-\Delta t} \end{Bmatrix} + \begin{Bmatrix} X_{t-2\Delta t} \\ Y_{t-2\Delta t} \\ Z_{t-2\Delta t} \end{Bmatrix} \right]$$

$$\begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} = \frac{4}{5} \left[\frac{\Delta t}{2} \begin{Bmatrix} \dot{p} + \dot{\phi}_{t-\Delta t} \\ \dot{\epsilon} + \dot{\theta}_{t-\Delta t} \\ \dot{\psi} + \dot{\psi}_{t-\Delta t} \end{Bmatrix} + \begin{Bmatrix} \phi_{t-2\Delta t} \\ \theta_{t-2\Delta t} \\ \psi_{t-2\Delta t} \end{Bmatrix} \right]$$

Subroutine

$$\begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{Bmatrix} X_{io} \\ Y_{io} \\ Z_{io} \end{Bmatrix} + \begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \phi_o \\ \theta_o \\ \psi_o \end{Bmatrix} + \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix}$$

SUMICS

$$-180^\circ \leq \psi \leq 180^\circ$$

$$\begin{Bmatrix} \phi_R \\ \theta_R \\ \psi_R \end{Bmatrix} = \frac{\pi}{180} \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}$$

SUMICS

Gravity Forces and Moments GRAVITY

Sum Forces and Moments FORCES

Equations of Motion MOTION

$$\begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} = \frac{4}{5} \left[\frac{\Delta t}{2} \begin{Bmatrix} \dot u + \dot u_{t-\Delta t} \\ \dot v + \dot v_{t-\Delta t} \\ \dot w + \dot w_{t-\Delta t} \end{Bmatrix} + \begin{Bmatrix} u_{t-2\Delta t} \\ v_{t-2\Delta t} \\ w_{t-2\Delta t} \end{Bmatrix} \right]$$

$$\begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix} = \frac{4}{5} \left[\frac{\Delta t}{2} \begin{Bmatrix} \dot p + \dot p_{t-\Delta t} \\ q + \dot q_{t-\Delta t} \\ \dot r + r_{t-\Delta t} \end{Bmatrix} + \begin{Bmatrix} p_{t-2\Delta t} \\ q_{t-2\Delta t} \\ r_{t-2\Delta t} \end{Bmatrix} \right]$$

Subroutine

$$\left\{\begin{matrix} u \\ v \\ w \end{matrix}\right\} = \left\{\begin{matrix} u_o \\ v_o \\ w_o \end{matrix}\right\} + \left\{\begin{matrix} u' \\ v' \\ w' \end{matrix}\right\}$$

SUMICS

$$\left\{\begin{matrix} p \\ q \\ r \end{matrix}\right\} = \left\{\begin{matrix} p_o \\ q_o \\ r_o \end{matrix}\right\} + \left\{\begin{matrix} p' \\ q' \\ r' \end{matrix}\right\}$$

SUMICS

$$\left\{\begin{matrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{matrix}\right\} = C_i^v \left\{\begin{matrix} u \\ v \\ w \end{matrix}\right\}$$

CTHREE

$$\left\{\begin{matrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{matrix}\right\} = \left\{\begin{matrix} p + \theta r \\ q - \phi r \\ \phi q + r \end{matrix}\right\}$$

CTHREE

$$\left\{\begin{matrix} u_{t-\Delta t} \\ v_{t-\Delta t} \\ w_{t-\Delta t} \end{matrix}\right\} = \left\{\begin{matrix} u' \\ v' \\ w' \end{matrix}\right\}$$

$$\left\{\begin{matrix} p_{t-\Delta t} \\ q_{t-\Delta t} \\ r_{t-\Delta t} \end{matrix}\right\} = \left\{\begin{matrix} p' \\ q' \\ r' \end{matrix}\right\}$$

$$\left\{\begin{matrix}\dot{u}_{t-\Delta t}\\ \dot{v}_{t-\Delta t}\\ \dot{w}_{t-\Delta t}\end{matrix}\right\} = \left\{\begin{matrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{matrix}\right\}$$

$$\left\{\begin{matrix}\dot{p}_{t-\Delta t}\\ \dot{q}_{t-\Delta t}\\ \dot{r}_{t-\Delta t}\end{matrix}\right\} = \left\{\begin{matrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{matrix}\right\}$$

$$\left\{\begin{matrix}X_{t-\Delta t}\\ Y_{t-\Delta t}\\ Z_{t-\Delta t}\end{matrix}\right\} = \left\{\begin{matrix}X_i'\\ Y_i'\\ Z_i'\end{matrix}\right\}$$

$$\left\{\begin{matrix}\Phi_{t-\Delta t}\\ \theta_{t-\Delta t}\\ \psi_{t-\Delta t}\end{matrix}\right\} = \left\{\begin{matrix}\Phi\\ \theta\\ \psi\end{matrix}\right\}$$

$$\left\{\begin{matrix} \dot{X}_{t-\Delta t} \\ Y_{t-\Delta t} \\ \dot{Z}_{t-\Delta t} \end{matrix}\right\} = \left\{\begin{matrix} X_i \\ \dot{Y}_i \\ \dot{Z}_i \end{matrix}\right\}$$

$$\left\{\begin{matrix} \dot{\phi}_{t-\Delta t} \\ \dot{\theta}_{t-\Delta t} \\ \dot{\psi}_{t-\Delta t} \end{matrix}\right\} = \left\{\begin{matrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{matrix}\right\}$$

For $t = n\Delta t,\ n > 1$

$$\begin{Bmatrix} P_x \\ P_y \\ P_z \end{Bmatrix} = \begin{Bmatrix} X_{t-2\Delta t} \\ Y_{t-2\Delta t} \\ Z_{t-2\Delta t} \end{Bmatrix} + 2\Delta t \begin{Bmatrix} \dot{X}_{t-\Delta t} \\ \dot{Y}_{t-\Delta t} \\ \dot{Z}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} P_\phi \\ P_\theta \\ P_\psi \end{Bmatrix} = \begin{Bmatrix} \phi_{t-2\Delta t} \\ \theta_{t-2\Delta t} \\ \psi_{t-2\Delta t} \end{Bmatrix} + 2\Delta t \begin{Bmatrix} \dot{\phi}_{t-\Delta t} \\ \dot{\theta}_{t-\Delta t} \\ \dot{\psi}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} P_u \\ P_v \\ P_w \end{Bmatrix} = \begin{Bmatrix} u_{t-2\Delta t} \\ v_{t-2\Delta t} \\ w_{t-2\Delta t} \end{Bmatrix} + 2\Delta t \begin{Bmatrix} \dot{u}_{t-\Delta t} \\ \dot{v}_{t-\Delta t} \\ \dot{w}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} P_p \\ P_q \\ P_r \end{Bmatrix} = \begin{Bmatrix} p_{t-2\Delta t} \\ q_{t-2\Delta t} \\ r_{t-2\Delta t} \end{Bmatrix} + 2\Delta t \begin{Bmatrix} \dot{p}_{t-\Delta t} \\ \dot{q}_{t-\Delta t} \\ \dot{r}_{t-\Delta t} \end{Bmatrix}$$

Subroutine

$$\begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix} = \begin{Bmatrix} P_x \\ P_y \\ P_z \end{Bmatrix} + \begin{Bmatrix} \Delta PC_x \\ \Delta PC_y \\ \Delta PC_z \end{Bmatrix}$$

$$\begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} = \begin{Bmatrix} P_\phi \\ P_\theta \\ P_\psi \end{Bmatrix} + \begin{Bmatrix} \Delta PC_\phi \\ \Delta PC_\theta \\ \Delta PC_\psi \end{Bmatrix}$$

$$\begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} = \begin{Bmatrix} P_u \\ P_v \\ P_w \end{Bmatrix} + \begin{Bmatrix} \Delta PC_u \\ \Delta PC_v \\ \Delta PC_w \end{Bmatrix}$$

$$\begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix} = \begin{Bmatrix} P_p \\ P_q \\ P_r \end{Bmatrix} + \begin{Bmatrix} \Delta PC_p \\ \Delta PC_q \\ \Delta PC_r \end{Bmatrix}$$

$$\begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{Bmatrix} X_{io} \\ Y_{io} \\ Z_{io} \end{Bmatrix} + \begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \phi_o \\ \theta_o \\ \psi_o \end{Bmatrix} + \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} \quad -180^\circ \leq \Psi \leq 180^\circ$$

SUMICS

Subroutine

$$\begin{Bmatrix} \phi_R \\ \theta_R \\ \psi_R \end{Bmatrix} = \frac{\pi}{180} \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u_o \\ v_o \\ w_o \end{Bmatrix} + \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} p_o \\ q_o \\ r_o \end{Bmatrix} + \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} \dot{X}_i \\ \dot{Y}_i \\ Z_i \end{Bmatrix} = C_i^v \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

CTHREE

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{Bmatrix} p + \theta r \\ q - \phi r \\ \phi q + r \end{Bmatrix}$$

CTHREE

$$\begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} = \begin{Bmatrix} X_{t-\Delta t} \\ Y_{t-\Delta t} \\ Z_t \end{Bmatrix} + \frac{\Delta t}{2} \begin{Bmatrix} \dot{X}_i + \dot{X}_{t-\Delta t} \\ \dot{Y}_i + \dot{Y}_{t-\Delta t} \\ \dot{Z}_i + \dot{Z}_{t-\Delta t} \end{Bmatrix}$$

Subroutine

$$\begin{Bmatrix} C_\phi \\ C_\theta \\ C_\psi \end{Bmatrix} = \begin{Bmatrix} \phi_{t-\Delta t} \\ \theta_{t-\Delta t} \\ \psi_{t-\Delta t} \end{Bmatrix} + \frac{\Delta t}{2} \begin{Bmatrix} \dot{\phi}+\dot{\phi}_{t-\Delta t} \\ \dot{\vartheta}+\dot{\theta}_{t-\Delta t} \\ \dot{\psi}+\dot{\psi}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta PC_x \\ \Delta PC_y \\ \Delta PC_z \end{Bmatrix} = \frac{4}{5} \begin{Bmatrix} C_x - P_x \\ C_y - P_y \\ C_z - P_z \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta PC_\phi \\ \Delta PC_\theta \\ \Delta PC_\psi \end{Bmatrix} = \frac{4}{5} \begin{Bmatrix} C_\phi - P_\phi \\ C_\theta - P_\theta \\ C_\psi - P_\psi \end{Bmatrix}$$

$$\begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix} = \begin{Bmatrix} P_x \\ P_y \\ P_z \end{Bmatrix} + \begin{Bmatrix} \Delta PC_x \\ \Delta PC_y \\ \Delta PC_z \end{Bmatrix}$$

$$\begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} = \begin{Bmatrix} P_\phi \\ P_\theta \\ P\psi \end{Bmatrix} + \begin{Bmatrix} \Delta PC_\phi \\ \Delta PC_\theta \\ \Delta PC_\psi \end{Bmatrix}$$

$$\begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{Bmatrix} X_{io} \\ Y_{io} \\ Z_{io} \end{Bmatrix} + \begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix}$$

SUMICS

Subroutine

$$\begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \phi_o \\ \theta_o \\ \psi_o \end{Bmatrix} + \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix} \quad -180^\circ \leq \psi \leq 180^\circ$$

SUMICS

$$\begin{Bmatrix} \phi_R \\ \theta_R \\ \psi_R \end{Bmatrix} = \frac{\pi}{180} \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}$$

SUMICS

$$\begin{Bmatrix} X_{t-2\Delta t} \\ Y_{t-2\Delta t} \\ Z_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} X_{t-\Delta t} \\ Y_{t-\Delta t} \\ Z_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} \phi_{t-2\Delta t} \\ \theta_{t-2\Delta t} \\ \psi_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} \phi_{t-\Delta t} \\ \theta_{t-\Delta t} \\ \psi_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} X_{t-\Delta t} \\ Y_{t-\Delta t} \\ Z_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} X_i' \\ Y_i' \\ Z_i' \end{Bmatrix}$$

$$\begin{Bmatrix} \phi_{t-\Delta t} \\ \theta_{t-\Delta t} \\ \psi_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} \phi' \\ \theta' \\ \psi' \end{Bmatrix}$$

	Subroutine
Gravity Forces and Moments	GRAVITY
Sum Forces and Moments	FORCES
Equations of Motion	MOTION

$$\begin{Bmatrix} C_u \\ C_v \\ C_w \end{Bmatrix} = \begin{Bmatrix} u_{t-\Delta t} \\ v_{t-\Delta t} \\ w_{t-\Delta t} \end{Bmatrix} + \frac{\Delta t}{2} \begin{Bmatrix} \dot{u} + \dot{u}_{t-\Delta t} \\ \dot{v} + \dot{v}_{t-\Delta t} \\ \dot{w} + \dot{w}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} C_p \\ C_q \\ C_r \end{Bmatrix} = \begin{Bmatrix} p_{t-\Delta t} \\ q_{t-\Delta t} \\ r_{t-\Delta t} \end{Bmatrix} + \frac{\Delta t}{2} \begin{Bmatrix} \dot{p} + \dot{p}_{t-\Delta t} \\ \dot{q} + \dot{q}_{t-\Delta t} \\ \dot{r} + \dot{r}_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta PC_u \\ \Delta PC_v \\ \Delta PC_w \end{Bmatrix} = \frac{4}{5} \begin{Bmatrix} C_u - P_u \\ C_v - P_v \\ C_w - P_w \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta PC_p \\ \Delta PC_q \\ \Delta PC_r \end{Bmatrix} = \frac{4}{5} \begin{Bmatrix} C_p - P_p \\ C_q - P_q \\ C_r - P_r \end{Bmatrix}$$

$$\begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} = \begin{Bmatrix} P_u \\ P_v \\ P_w \end{Bmatrix} + \begin{Bmatrix} \Delta PC_u \\ \Delta PC_v \\ \Delta PC_w \end{Bmatrix}$$

Subroutine

$$\left\{\begin{matrix} p' \\ q' \\ r' \end{matrix}\right\} = \left\{\begin{matrix} P_p \\ P_q \\ P_r \end{matrix}\right\} + \left\{\begin{matrix} \Delta PC_p \\ \Delta PC_q \\ \Delta PC_r \end{matrix}\right\}$$

$$\left\{\begin{matrix} u \\ v \\ w \end{matrix}\right\} = \left\{\begin{matrix} u_o \\ v_o \\ w_o \end{matrix}\right\} + \left\{\begin{matrix} u' \\ v' \\ w' \end{matrix}\right\}$$

SUMICS

$$\left\{\begin{matrix} p \\ q \\ r \end{matrix}\right\} = \left\{\begin{matrix} p_o \\ q_o \\ r_o \end{matrix}\right\} + \left\{\begin{matrix} p' \\ q' \\ r' \end{matrix}\right\}$$

SUMICS

$$\left\{\begin{matrix} X_i \\ \dot{Y}_i \\ \dot{Z}_i \end{matrix}\right\} = C^v_i \left\{\begin{matrix} u \\ v \\ w \end{matrix}\right\}$$

CTHREE

$$\left\{\begin{matrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{matrix}\right\} = \left\{\begin{matrix} p + \theta r \\ q - \phi r \\ \phi q + r \end{matrix}\right\}$$

CTHREE

$$\left\{\begin{matrix} u_{t-2\Delta t} \\ v_{t-2\Delta t} \\ w_{t-2\Delta t} \end{matrix}\right\} = \left\{\begin{matrix} u_{t-\Delta t} \\ v_{t-\Delta t} \\ w_{t-\Delta t} \end{matrix}\right\}$$

$$\begin{Bmatrix} p_{t-2\Delta t} \\ q_{t-2\Delta t} \\ r_{t-2\Delta t} \end{Bmatrix} = \begin{Bmatrix} p_{t-\Delta t} \\ q_{t-\Delta t} \\ r_{t-\Delta t} \end{Bmatrix}$$

$$\begin{Bmatrix} u_{t-\Delta t} \\ v_{t-\Delta t} \\ w_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix}$$

$$\begin{Bmatrix} p_{t-\Delta t} \\ q_{t-\Delta t} \\ r_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} p' \\ q' \\ r' \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{u}_{t-\Delta t} \\ \dot{v}_{t-\Delta t} \\ \dot{w}_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{p}_{t-\Delta t} \\ \dot{q}_{t-\Delta t} \\ \dot{r}_{t-\Delta t} \end{Bmatrix} = \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix}$$

$$\left\{\begin{matrix}\dot{X}_{t-\Delta t}\\ \dot{Y}_{t-\Delta t}\\ \dot{Z}_{t-\Delta t}\end{matrix}\right\} = \left\{\begin{matrix}\dot{X}_i\\ \dot{Y}_i\\ \dot{Z}_i\end{matrix}\right\}$$

$$\left\{\begin{matrix}\dot{\phi}_{t-\Delta t}\\ \dot{\theta}_{t-\Delta t}\\ \dot{\psi}_{t-\Delta t}\end{matrix}\right\} = \left\{\begin{matrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{matrix}\right\}$$

TRANSCENDENTAL FUNCTIONS

The transcendental functions are the standard arithmetic functions that are implemented as library subroutines for general purpose usage.

ARCTANGENT. This function was taken directly from the Sigma 7 FORTRAN library (6ATAN). It is a two argument function. The one argument entry was removed. See Figure 33 for a computer generated plot.

COSINE. This function was taken directly from the Sigma 7 FORTRAN library (6COS). The sine entry was removed. See Figure 34 for a computer generated plot.

SINE AND COSINE. These functions were specially developed as curve fits for fast execution. The cosine is simply 90^o phase shifted from the sine with angle inputs limited to $\pm 360^o$. This function is a piecewise linear ($f(x) = A_iX + B_i$) fit of equal 10^o segments using a min-max (minimize the maximum error) fit and adjusted for function continuity at the segment boundaries. The maximum absolute error equals .0019 at $\pm 80^o$ and $\pm 110^c$. See Figures 35 through 40 for the functions, absolute error, and percentage error.

SQUARE ROOT. This function was taken directly from the Sigma 7 FORTRAN library (6SQRT).

Other function approximations were developed specially for fast execution. This function is a piecewise linear ($f(x) = A_iX + B_i$) fit with segment length chosen allowing a maximum error of .5% using a min-max fit. The maximum absolute error lies at the lower boundary. This function (7SGNSQR) is a signed square root:

$$f(x) = \sqrt{x}\,\frac{x}{|x|}$$

It is implemented as:

$$f(x) = \begin{cases} A_iX + B_i & X > 0 \\ A_iX - B_i & X \leq 0 \end{cases}$$

See Figure 41 for computer plot of function.

Another function (7ASQRT) is the same except that:

$$f(x) = X \qquad |X| \leq 1$$

See Figure 42 for computer plot of function.

DERIVATIVE OF SQUARE ROOT.

This function is a piecewise hyperbolic ($f(x) = A_i/X+B_i$) fit with segment length chosen allowing a maximum error of 2% using a min-max fit The maximum absolute error lies at the lower boundary. This function (7 DASQRT) is a signed square root derivative:

$$f(x) = \begin{cases} \dfrac{1}{2\sqrt{|X|}} & |X| > 1 \\ 1 & |X| \leq 1 \end{cases}$$

See Figure 43 for computer plot of function.

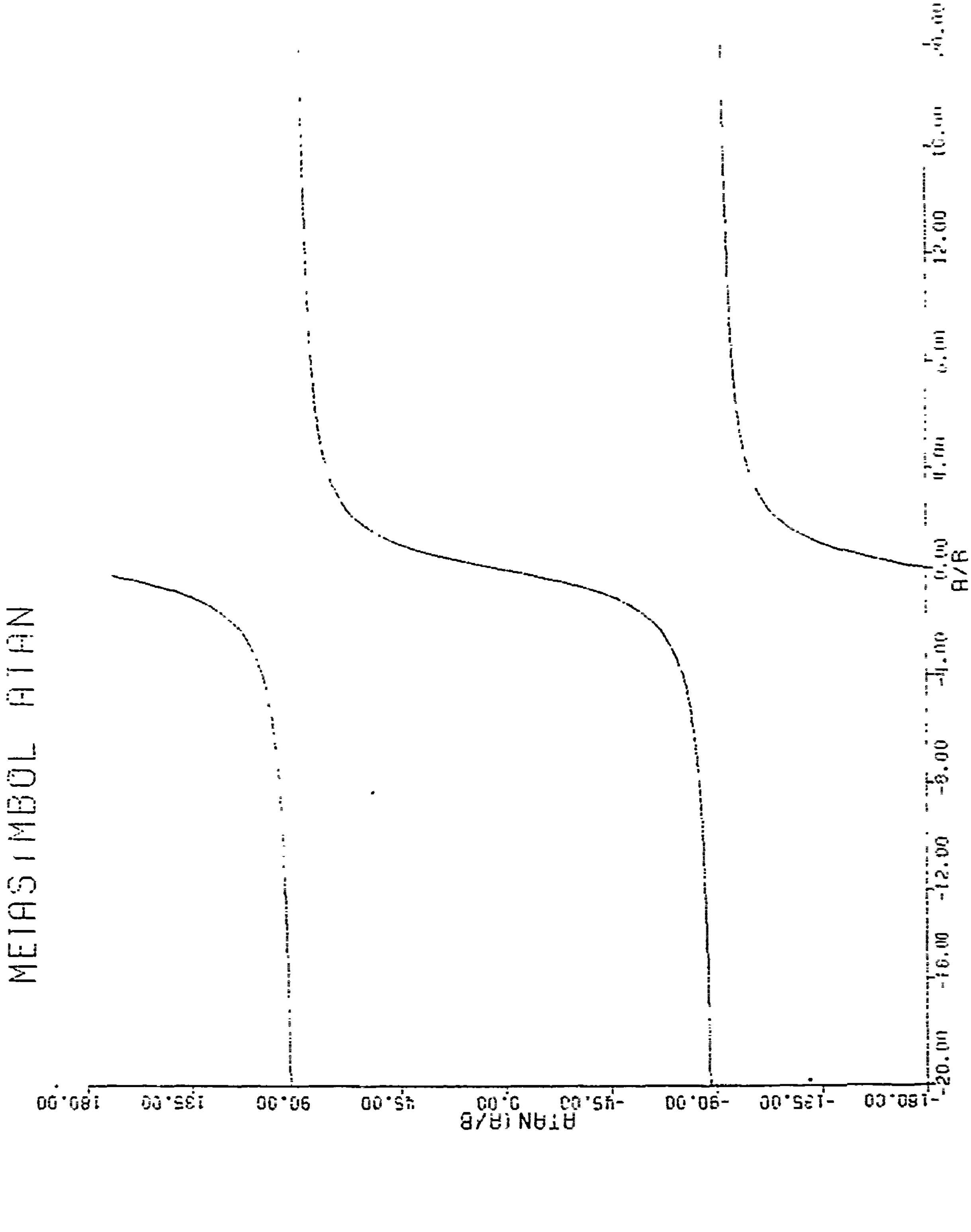


Figure 33. Arctangent

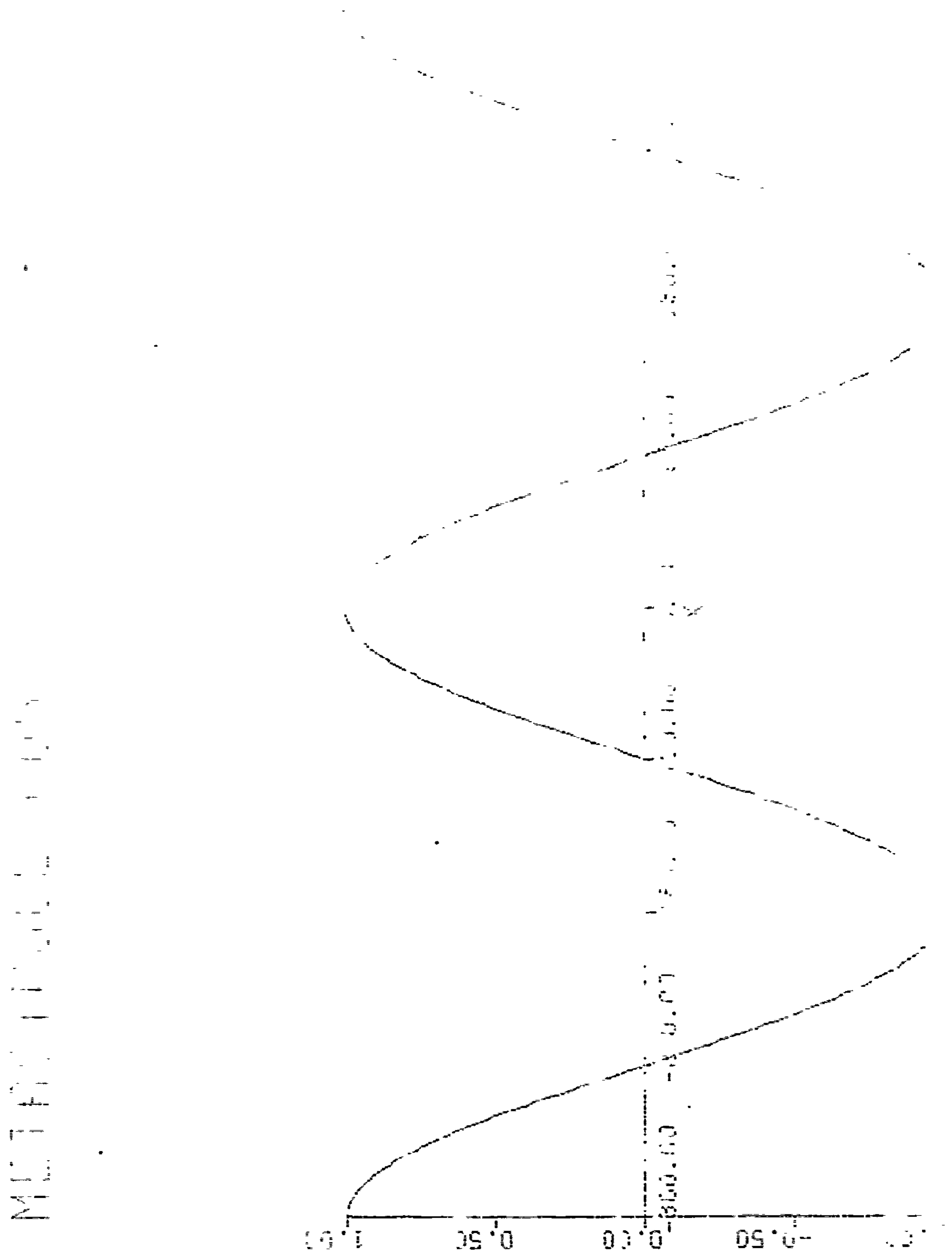


Figure 34. Cosine

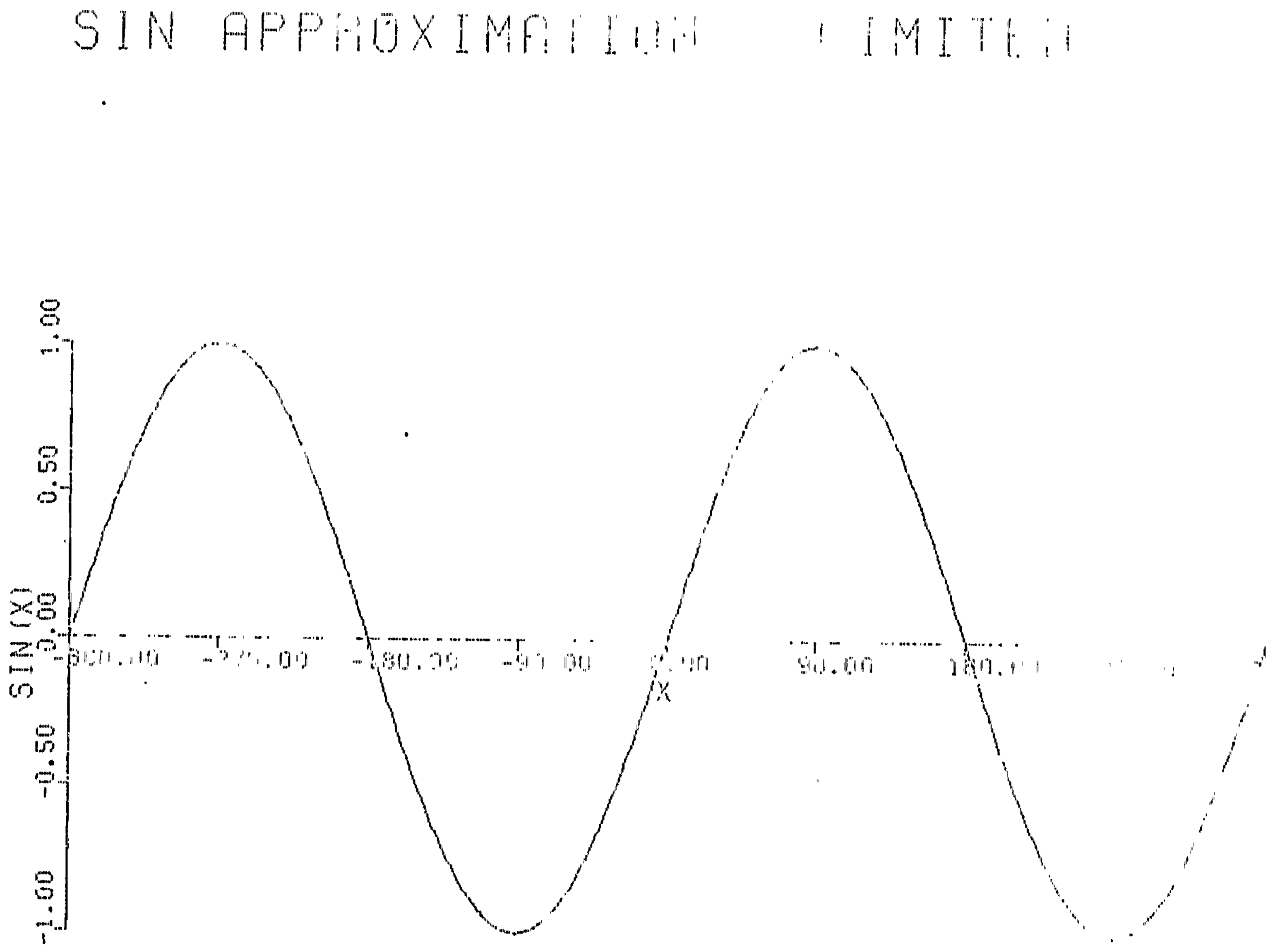


Figure 35. Sine Approximation

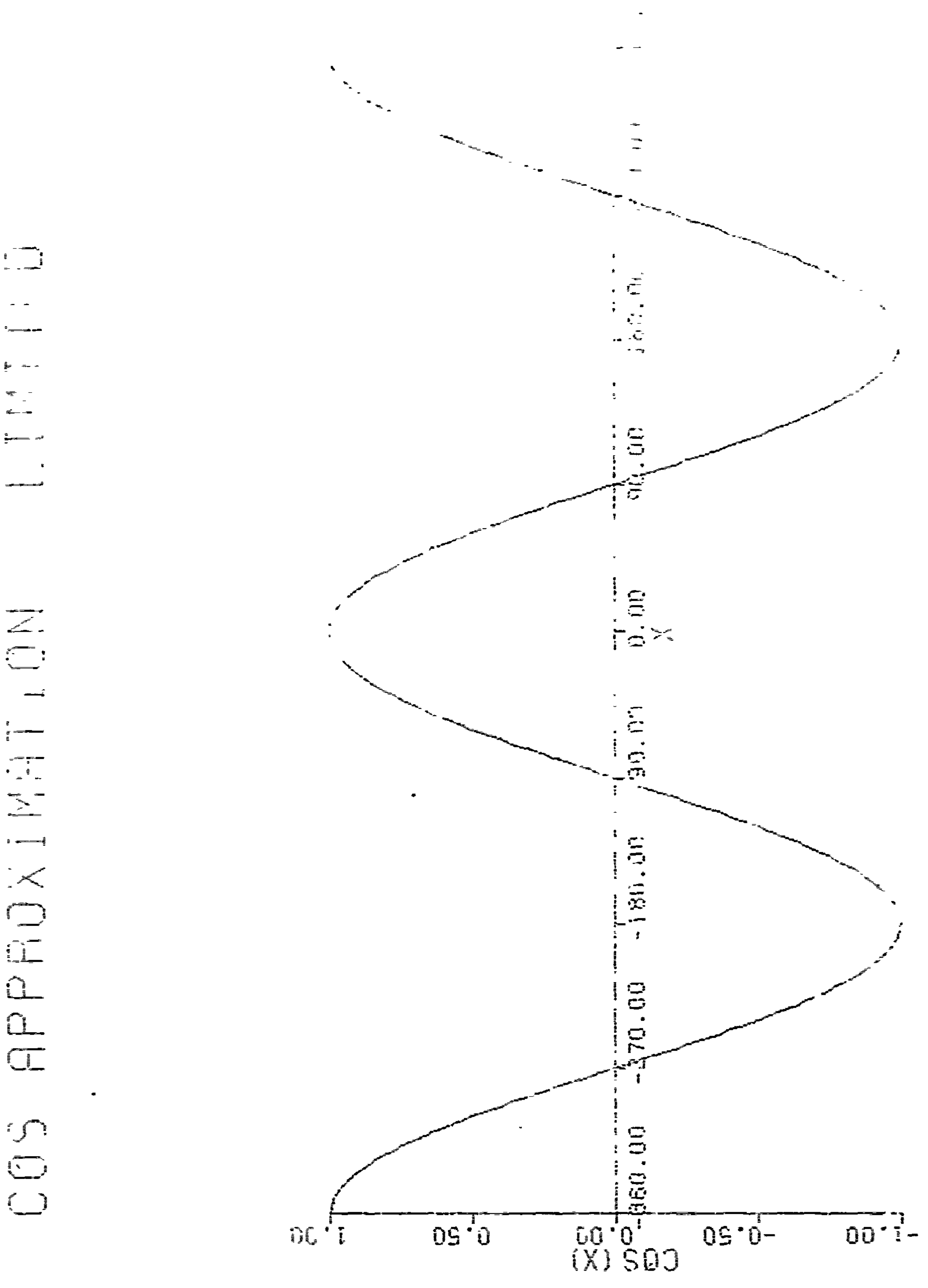


Figure 36. Cosine Approximation

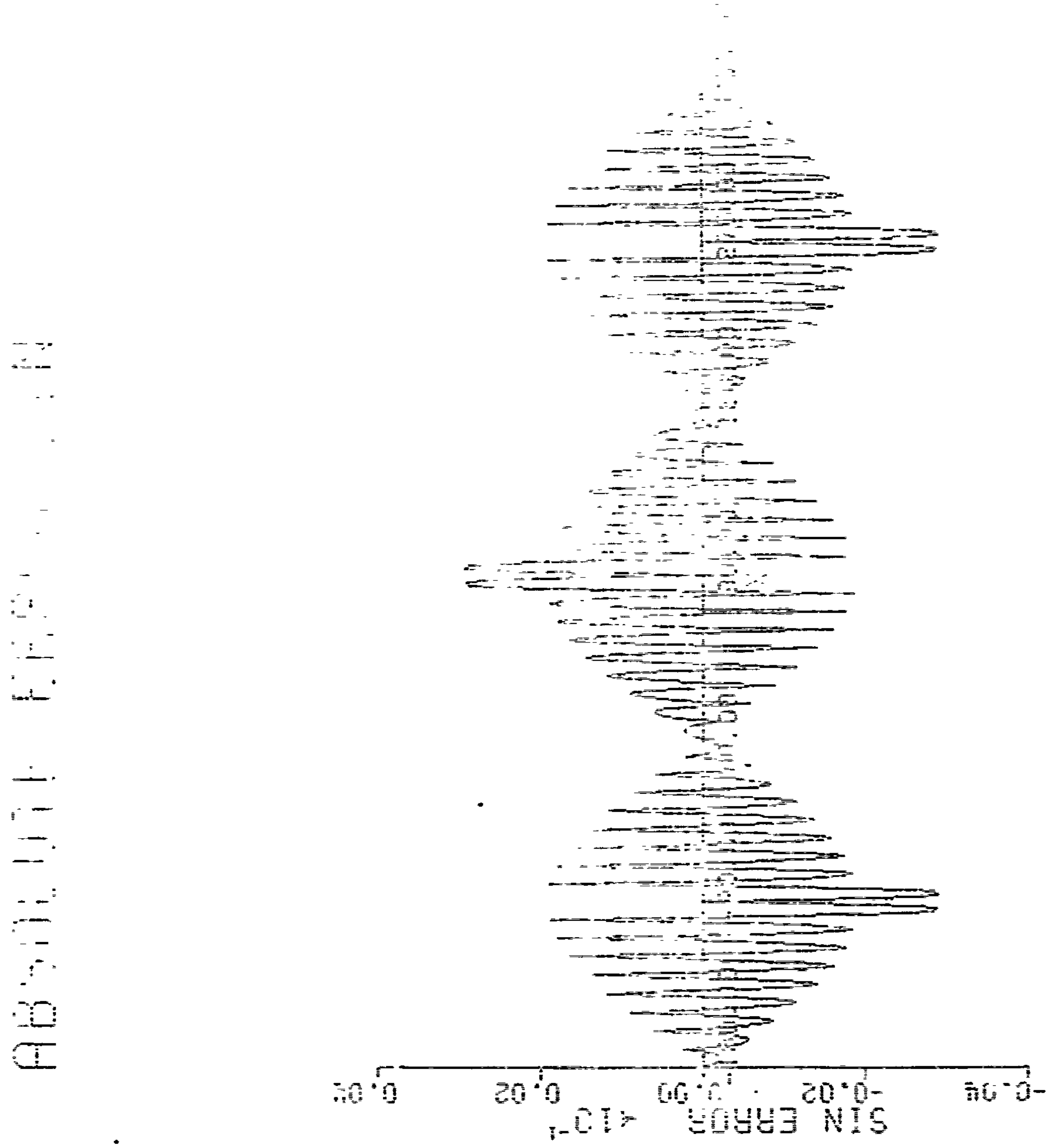


Figure 37. Sine Absolute Error

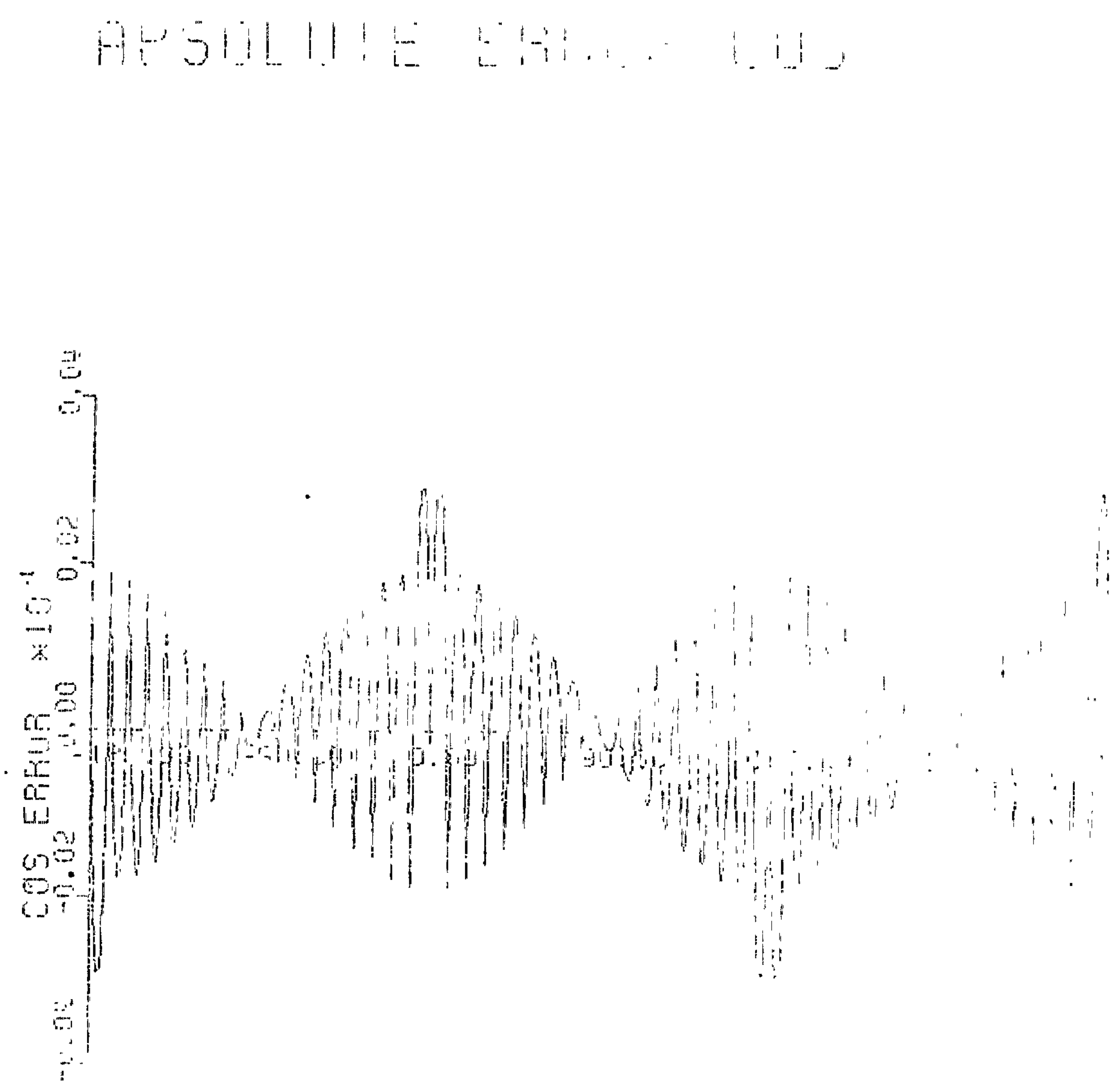


Figure 38. Cosine Absolute Error

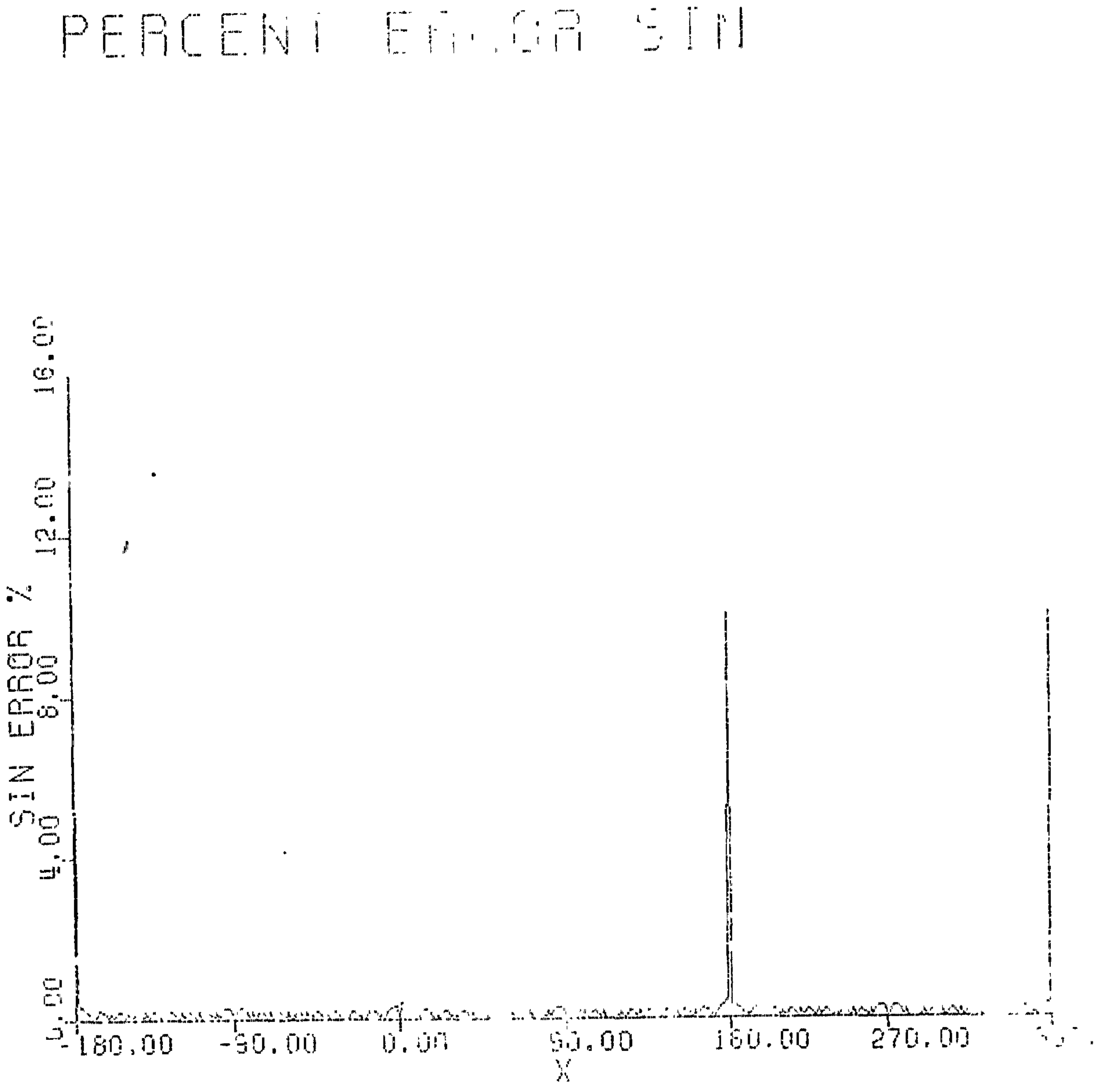


Figure 39. Sine Percentage Error

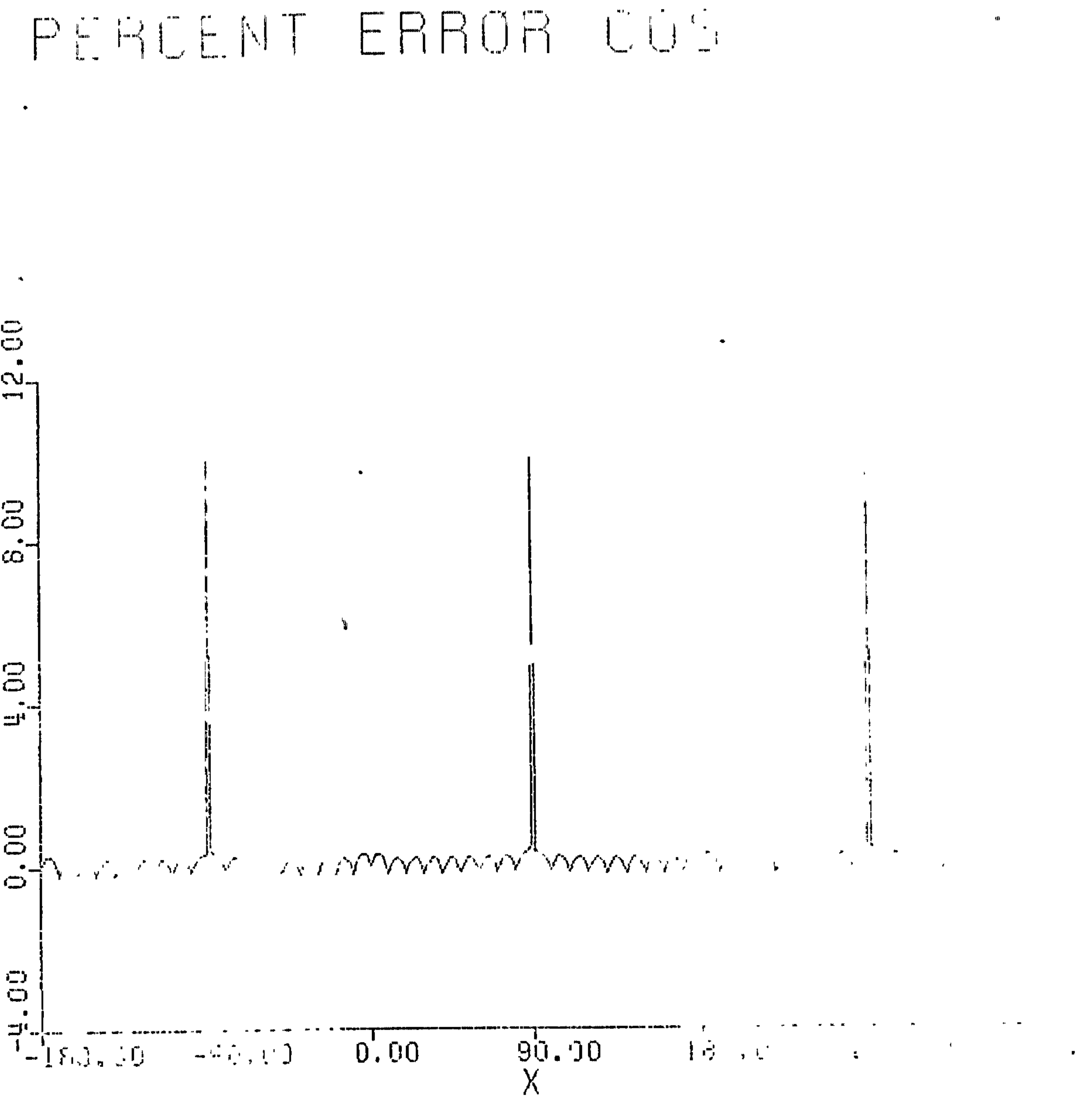


Figure 40. Cosine Percentage Error

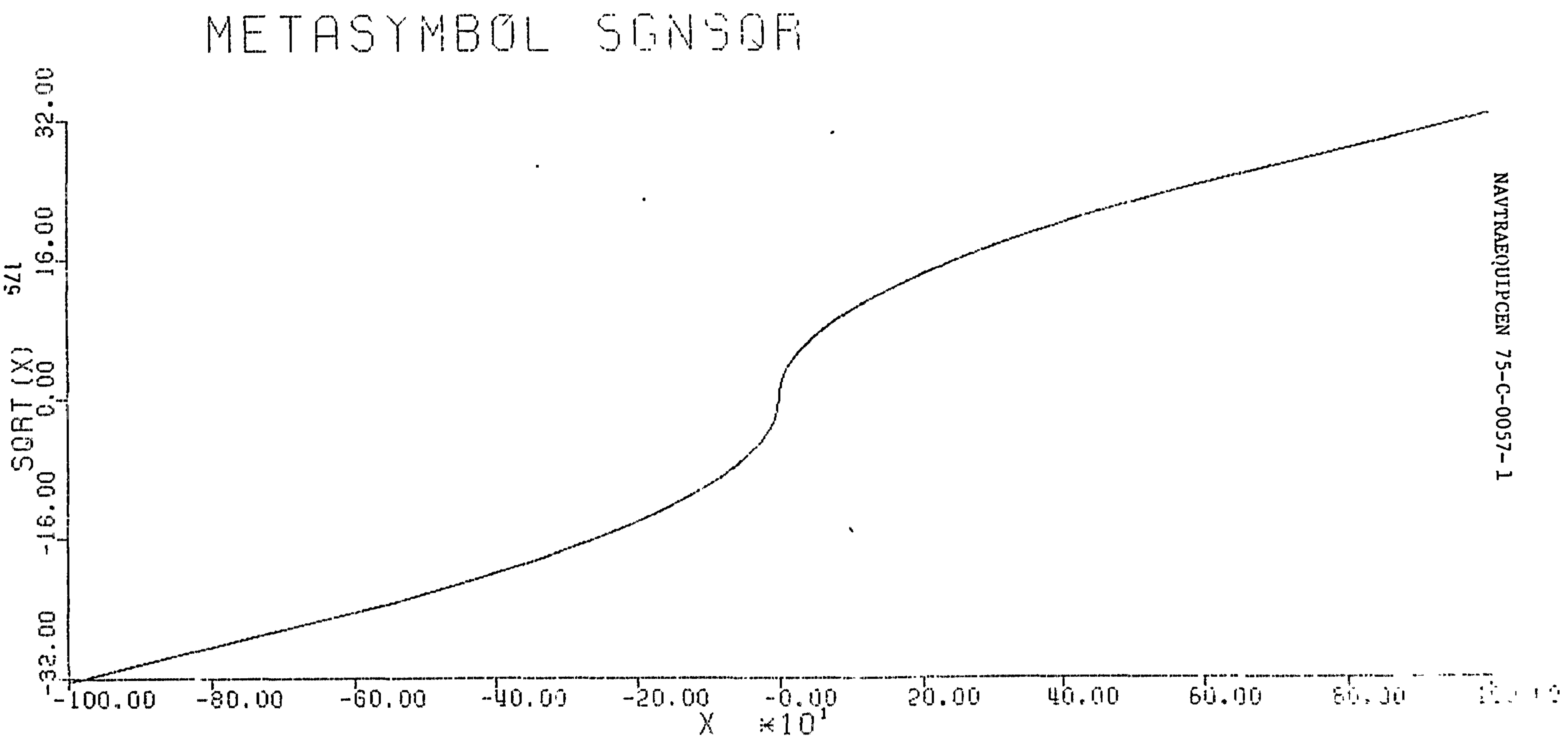


Figure 41. Signed Square Root Approximation

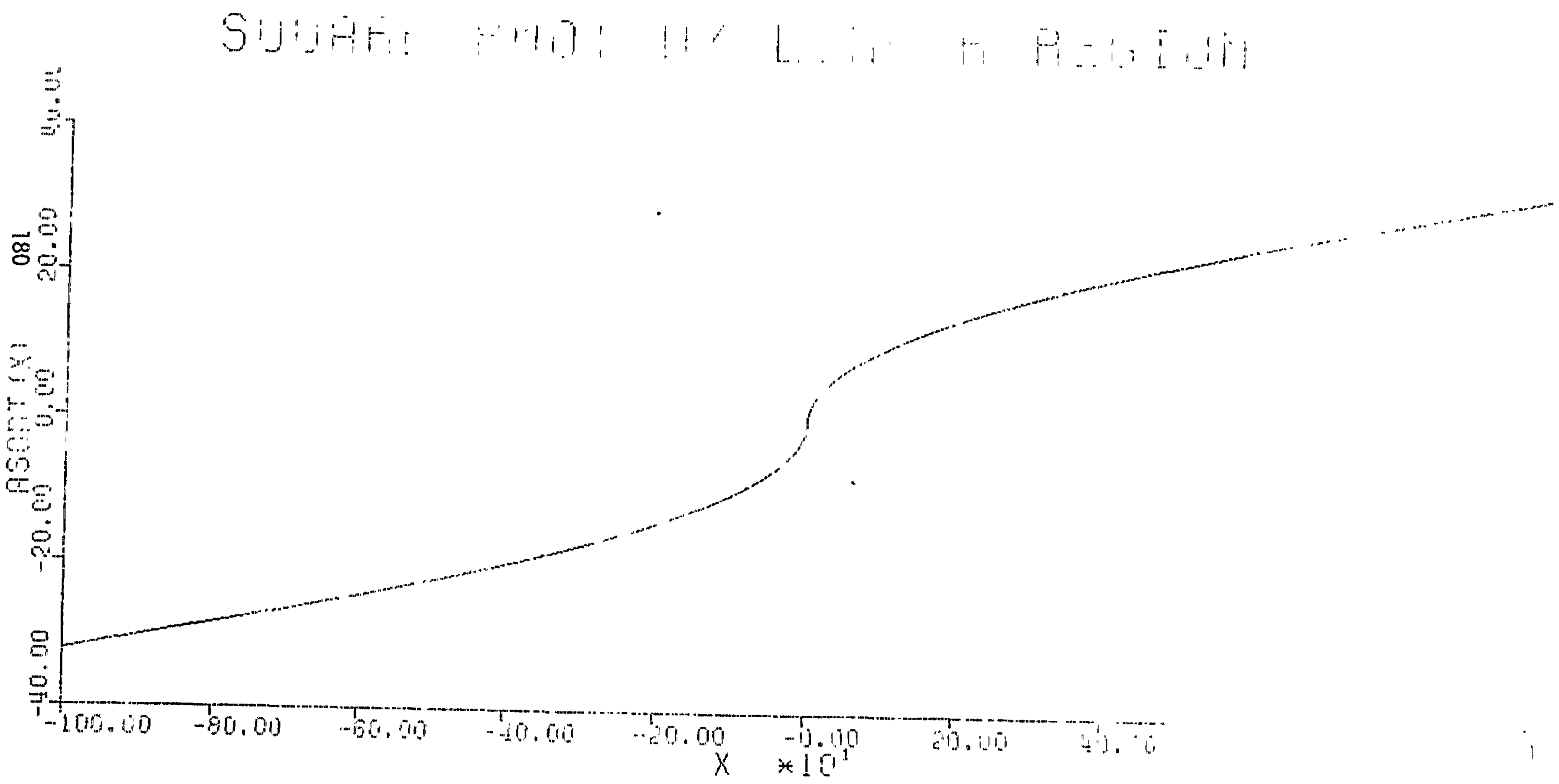


Figure 42 . Signed Square Root Approximation with Linear Region

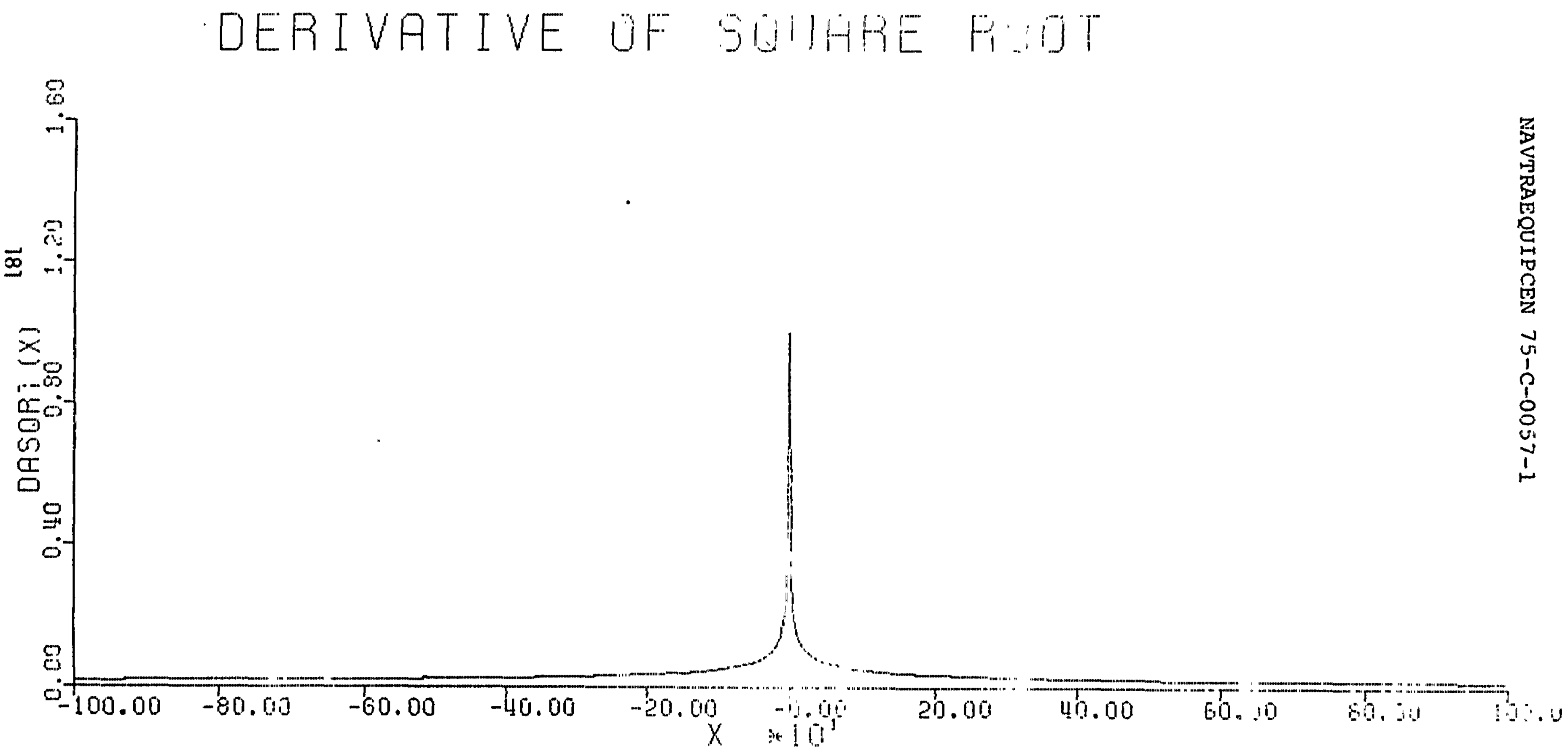


Figure 43 . Signed Square Root Derivative Approximation

CRT DOCKING AID-VISUAL

In order to give the pilot visual affects that would simulate those on the craft, a line drawing was created on a cathode ray tube (CRT). This display simulates a three dimensional picture of the mother ship's stern well to facilitate docking. Range and bearing and a target showing relative size according to distance are shown to the pilot.

No attempt is made here to describe the method or development of this visual simulation. The display was developed and programmed at NTEC and interfaced by CSDL to the real-time simulation.

Although an attempt to display a mother ship is made and docking maneuvers can be performed, no modelling of this complex two body problem has been considered.

Another Visual Simulation

As a part of the ACV-LC trainer, another visual simulation has also been installed. This visual simulation needs signals from the simulation. This part of the trainer was developed and interfaced to the real-time simulation by NAVTRAEQUIPCEN. No attempt at description is made here.

INITIALIZATION

In order to begin execution of the mathematical model equations, certain variables must be initialized. Since this model was developed around a quasi-operating point of the vehicle, no startup procedures have been defined. Initialization of the vehicle to this state must be performed. The state chosen for initialization is a hovering condition at zero speed with the vehicle supported by its air cushion.

The individual parameters to be initialized are all encompassing, but their values were chosen by trial and error. Other parameters that need to be initialized are calculated by their respective equations and by properly arranging the flow of execution.

The parameters which need to be initialized are simply the effectors and the pressures. This establishes the proper quasi-operating condition of the vehicle. Correspondingly, however, the pilot commands are initialized on a one to one basis with the actual control device.

Subroutine: SYSACVIC

$$N_{Tn} = N_{Tn\phi}$$

$$N_{Sj} = N_{Sj\phi}$$

$$N_{FANj} = N_{Sj}\, GR_{FAN}$$

$$N_{Pj} = N_{FANj} GR_{p}$$

$$P_{CUSH\ i} = P_{CUSH\ i\phi}$$

$$\beta_{Pj} = \beta_{Pj\phi}$$

$$P_{MANj} = P_{MANj\phi}$$

$$\psi_{N} = \psi_{N\phi}$$

j	1, n=1, 2, 3 Starboard
	2, n=4, 5, 6 Port
i=1	STBD Fwd cushion compartment
i=2	STBD Aft cushion compartment
i=3	PORT AFT cushion compartment
i=4	PORT FWD cushion compartment
$P_{CUSH\ i}$	Pressure within cushion compartment (Lbs/Ft^2)
$P_{CUSH\ i\phi}$	Initial value of $P_{CUSH\ i}$ = 109.37297 lbs/ft^2
N_{Tn}	Turbine speed (RPM)
$N_{Tn\phi}$	Initial turbine speed = 16,800 RPM
N_{Sj}	Power shaft speed (RPM)
$N_{Sj\phi}$	Initial power shaft speed = 13,200 RPM
N_{FANj}	Fan speed (RPM)
N_{Pj}	Propeller speed (RPM)
β_{Pj}	Pitch angle of propeller (Deg)
$\beta_{Pj\phi}$	Initial pitch angle of propeller = 12.9°
P_{MANj}	Manifold pressure (Lbs/Ft^2)
$P_{MANj\phi}$	Initial manifold pressure = 130.9 lbs/ft^2
ψ_N	Thrust nozzle angle (Deg)
ψ_N	Initial thrust nozzle angle = 180°
GR_{FAN}	Power shaft to fan gear ratio= .1297
GR_P	Fan to propeller gear ratio = .6427

Subroutine: CMDINIT

$\beta_{PCW} = 0$

$\psi_{NC} = 0$

$\psi_{RC} = 0$

$\psi_{NS} = 0$

$\beta_{PjCL} = \beta_{Pj\phi}$

$N_{TCn} = N_{Tn\phi}$

$N_{SCj} = N_{Sj\phi}$

j 1, n=1, 2, 3 Starboard

2, n=4, 5, 6 Port

β_{PCW} Propeller pitch vernier wheel command (Deg)

ψ_{NC} Nozzle angle command (Deg)

ψ_{RC} Rudder angle command (Deg)

ψ_{NS} Nozzle angle switch (Fwd/Aft)

β_{PjCL} Propeller pitch angle command (Deg)

$\beta_{Pj\phi}$ Initial propeller pitch angle = 12.9^{o}

N_{TCn} Turbine speed command (RPM)

$N_{Tn\phi}$ Initial turbine speed = 16,800 RPM

N_{SCj} Power shaft speed command (RPM)

$N_{Sj\phi}$ Initial power shaft speed = 13,200 RPM

SECTION IV

SOFTWARE STRUCTURE

The software structure is a modular system and is basically comprised of 5 major divisions:

Program load and initialization

Background

Foreground

Trap handling

Common storage

The first three tasks are linked directly by program flow and shown in Figure 44 . Traps are handled for normal and abnormal occurrences in these three tasks. The separate common storage (module DATAPOOL) contains data (constants and variables) which is used or manipulated by the 4 above tasks.

LOAD AND INITIALIZATION

The ACV-LC real-time simulation program is loaded from 9 track magnetic tape. Once loaded program execution begins with all interrupts inhibited. The traps handled by the program are initialized and are:

Control panel

Counter 3 zero

Counter 3

Power up

Power down

CAL4

Memory parity.

Non-allowed operation

Unimplemented instruction

Push down stack

Fixed point arithmetic overflow

Floating point fault

Decimal arithmetic fault

Watch Dog timer runout

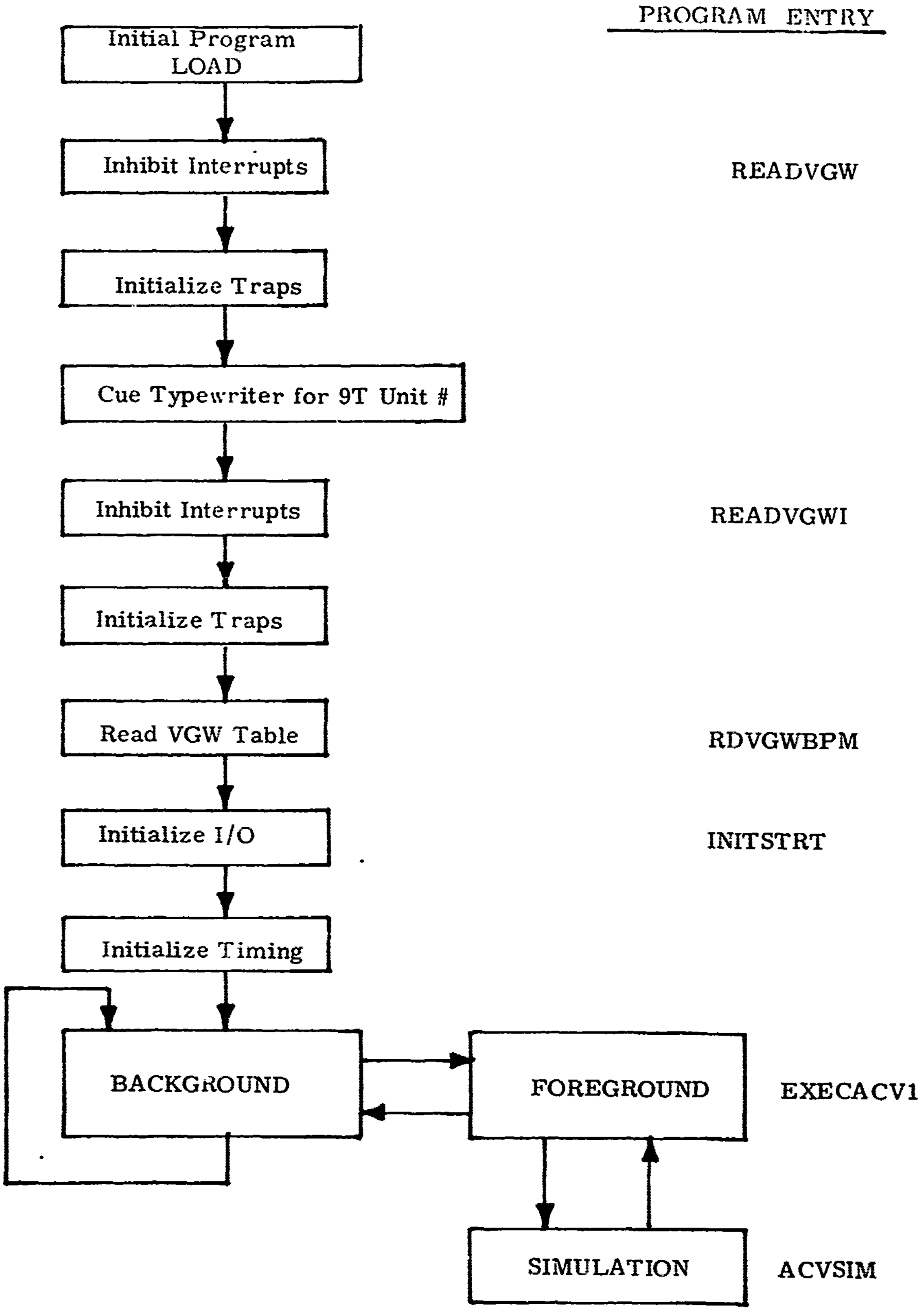


Figure 44 . Basic Software Flow and Structure

Restart

CAL1

The vehicle generated wave table of Kernel values is read if the option was selected at the time of load tape generation and if the proper response is made to typewriter cue. The table is read from the tape and stored in sequence. The beginning addresses of each subtable are stored in a separate table.

Initialization of all Input/Output (I/O) is performed. All I/O software request flags are reset. The time of day and the date are reset and the type-in flag is set to request the proper time and date. The I/O hardware and the audible alarm are reset to their usable states.

The program timing initialization is performed by resetting and disabling all interrupts and by setting the Counter 3 interrupt. The interrupts used are enabled:

Counter 3

Counter 3 Zero

Memory Parity

Control Panel

Program Entry

At the time of self loading tape creation, a transfer address is specified which will be the point of initial execution after program load. There are 3 beginning points from which the simulation program can start (see Figure 44):

READVGW

READVGWI

INITSTRT

READVGW

If READVGW is selected as the program entry, this message is typed on the console typewriter:

TYPE 9T UNIT NO. FOR VGW TABLE

OR NL TO NOT READ TABLE:

The typewriter input light is turned on and the program waits for an input by the operator. This input must be a decimal digit and/or a new line(NL) character.

If a decimal digit I and NL is typed, the vehicle generated wave kernel table will be read from 9 track (9T) magnetic tape (MT) unit number I. Transfer will be made to the program entry READVGWI.

If an NL character, only, is typed, the vehicle generated wave table will not be read and the program will transfer to the program entry INITSTRT.

READVGWI

If READVGWI is selected as the program entry, the 9T-MT unit number equals 0 and the vehicle generated wave kernel table will be read from the end of the self loading program tape.

When the kernel table is read successfully, these messages are typed to the operator:

```
VGW TABLE SUCCESSFULLY READ

NO. OF SUBTABLES READ = 82
```

The flag, VGWFLAG, will be set equal to 1 to run the VGW package and the program will transfer to the program entry INITSTRT.

INITSTRT

If the vehicle generated wave kernel table has not been read by the time this program entry has been reached, the VGW package will not be run in simulation. The program entry INITSTRT initializes the traps, I/O, and the timing and transfers to the Background Task (BACKGRND) for system operation.

BACKGROUND

The Background Task takes up all computer time not spent in the Foreground. The primary function of the Background is to perform I/O. The several functions are: typewriter input/output, card reader input, line printer output, and the CRT display. Other functions are also performed; the calculation of dwell time (amount of time spent in the background) and the vehicle generated wave height profile calculations.

Typewriter input allows the operator to change the contents of any memory location, the time of day, and the date. Typewriter output displays the contents of any memory location in any of several formats. The typewriter may also display software status messages to the operator. Cards may be read to perform modification to data within memory. The line printer may be cued to display the state of the ACV-LC in a clear format or the contents of the DATAPOOL may be dumped.

PROGRAMMING. Figures 45 through 49 show the flow of module execution of the Background Task. These flow diagrams are functional in nature and do not show the minute programming details. The major modules making up the Background Task are:

BACKGRND

TYPE

TYPEMSG

CARDS

CARDIO

PRINT

These modules are cycled indefinitely and are subject to interruption by traps, but primarily by the Foreground Task. Other modules are called as needed to perform specific functions (i.e., CARDREAD(M:READ), PRNTLINE(M:PRINT).

Dwell Time

The background timer is a counter that keeps track of the length of time that the background task is in operation. The value of the counter is saved and reset each cycle by the foreground. Subtracting the background time from the time of a complete ACV simulation sampling period will provide the time spent in the foreground. The timer (BGTIME) is incremented in units of 12 microseconds. It is saved in location BGTIMSAV. With the background timer active, the peripheral equipment and the CRT and the VGW cannot operate. The background timer occupies the entire time spent in the Background Task.

Switches

There are 2 sets of operator controlled switches which control the simulation program, the console sense switches (CSW) and the system switches (SSW). The switches and their functions in the Background are:

CSW 1	deactivate VGW & CRT
3	deactivate CRT display
SSW 8	Read cards
9	call SYSPRINT
11	dwell time
15	Call PRTDATA

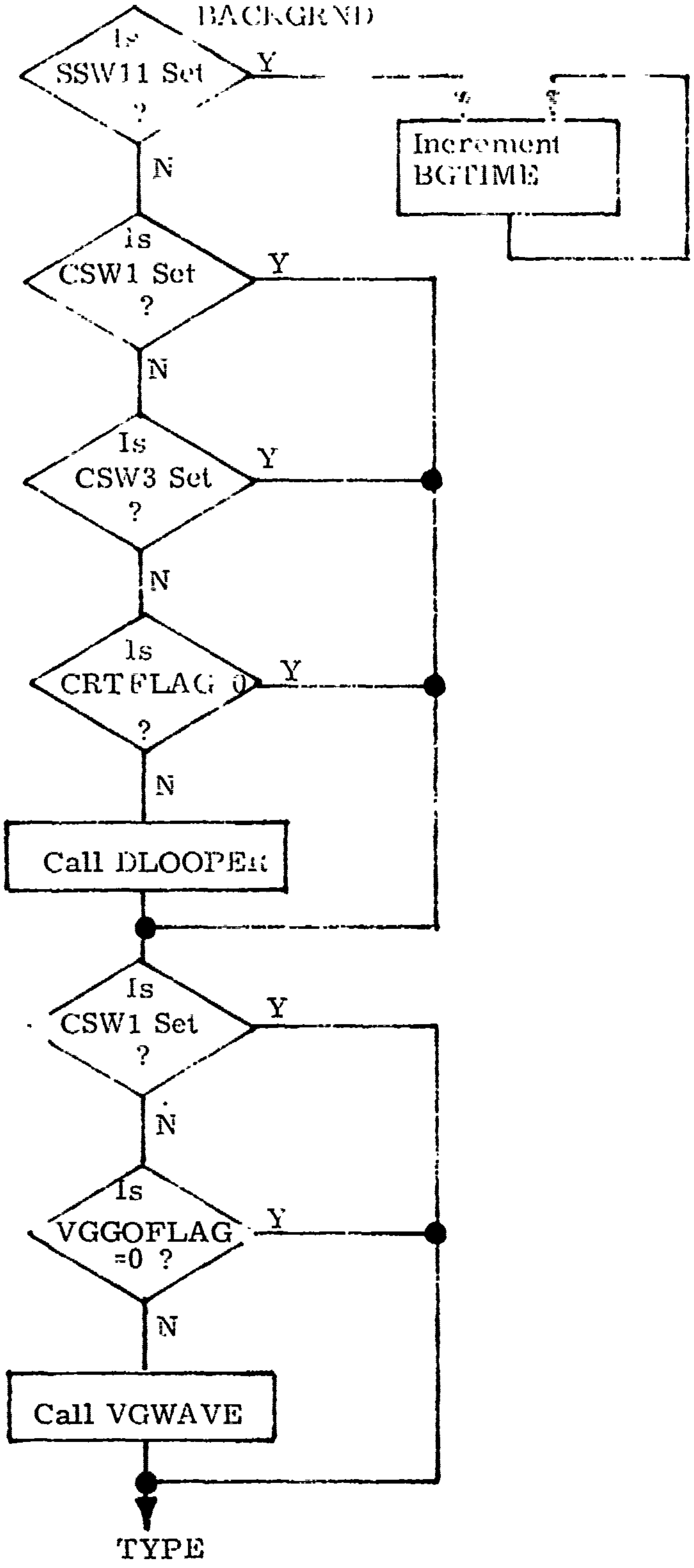


Figure 45. Background Task Flow-BACKGRND

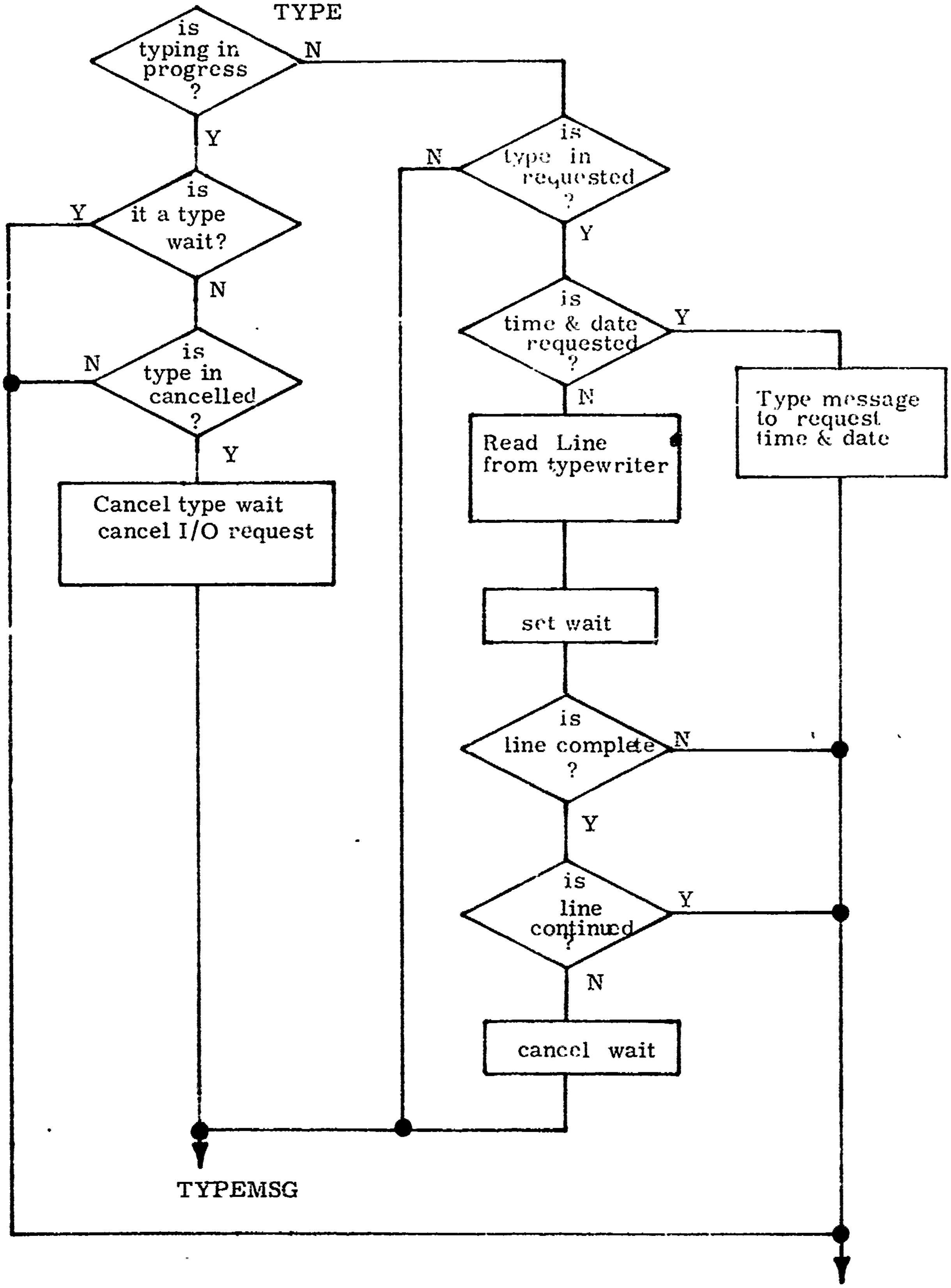


Figure 46. Background Task Flow - TYPE [illegible]

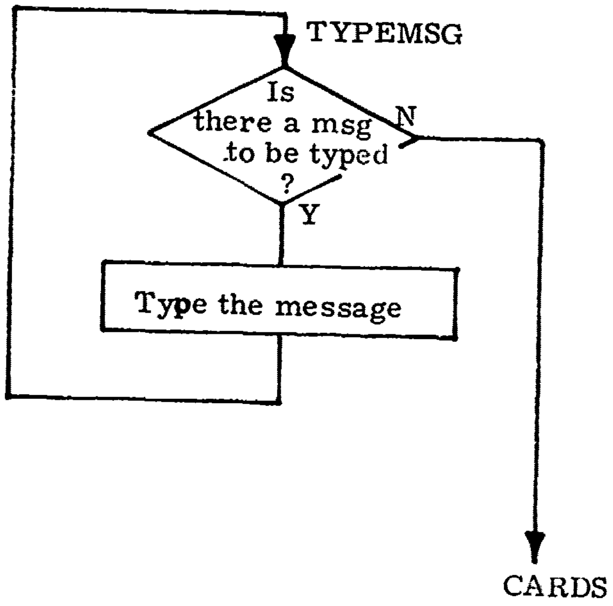


Figure 47 . Background Task Flow-TYPEMSG

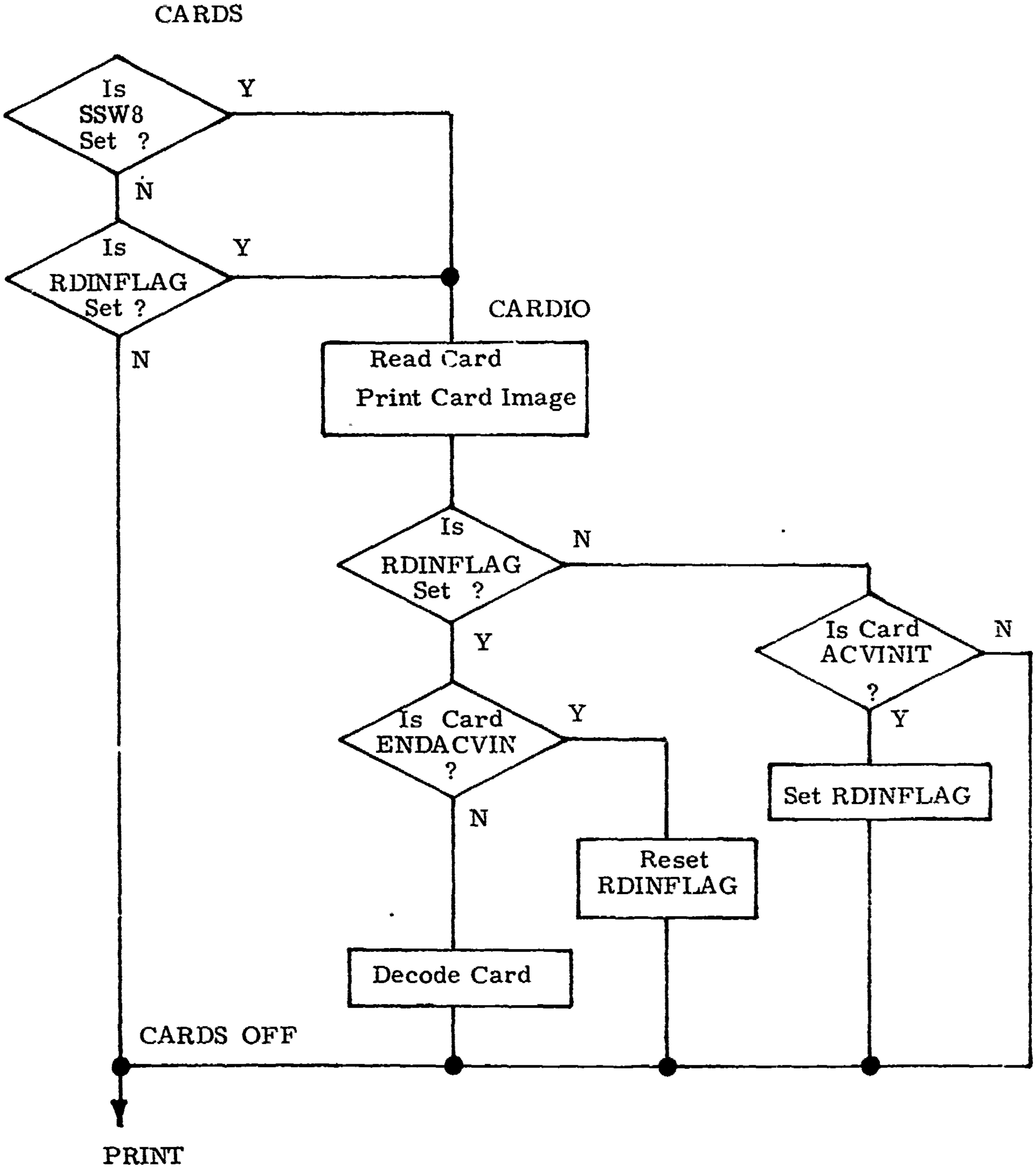


Figure 48. Background Task Flow-CARDS

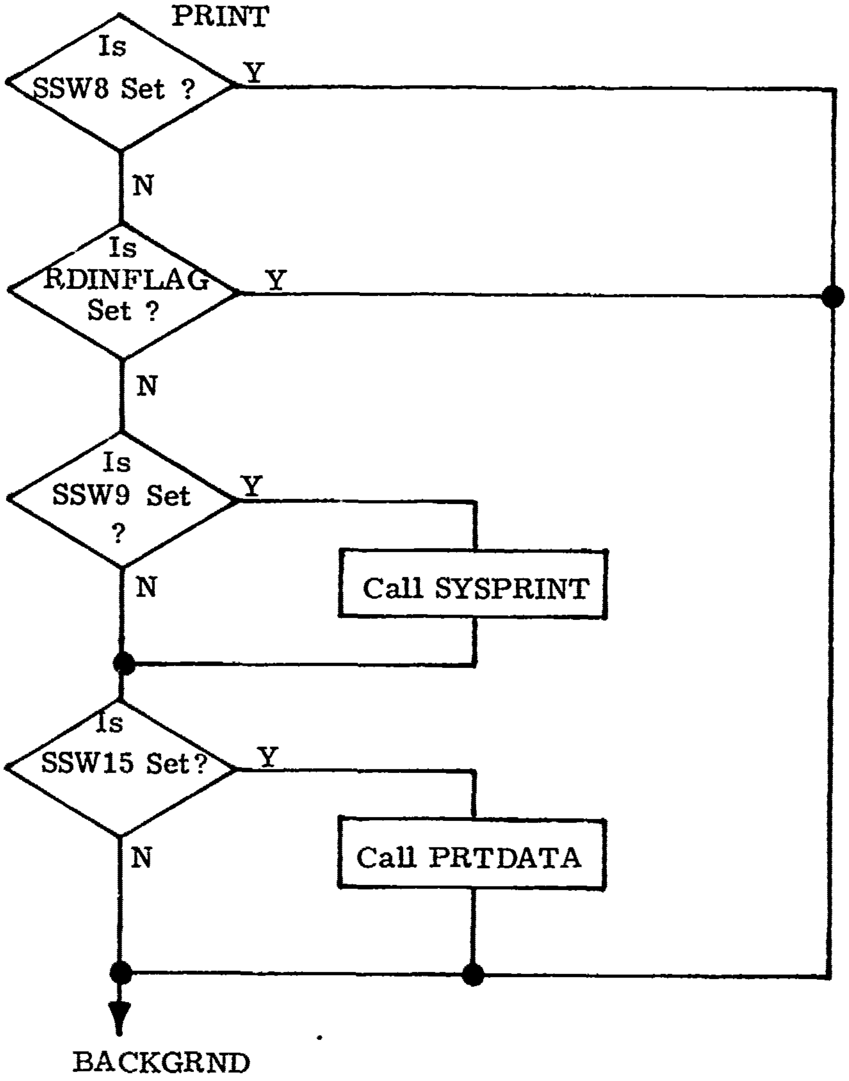


Figure 49. Background Task Flow-PRINT

SIMULATION RELATED FUNCTIONS. The specific modules which are shown are called to perform a particular background simulation function. These are:

DLOOPER

VGWAVE

SYSPRINT

PRTDATA

DLOOPER

This module does the necessary data conversion to be used for generation of the CRT display and calls DLOOP. DLOOP generates the CRT display.

VGWAVE

This module calculates the vehicle generated wave height at one point under the hull planform. Successive calls calculate the wave height at the next point.

SYSPRINT

This module prints the ACV state vectors, forces and moments, and the status of effectors at their current values. This printout is formatted for readability.

PRTDATA

This module dumps the DATAPOOL contents onto the line printer by hexidecimal address in floating point short (FS) format.

FLAGS

CRTFLAG is a software flag to activate the CRT display. This flag is set in the foreground.

VGGOFLAG is a software flag to execute the VGW profile equations. This flag is set in the foreground.

FOREGROUND

The Foreground Task is activated from the Counter 3 Zero interrupt at regular intervals. This task updates the time of day, saves the computed dwell time from the Background and then resets the dwell time counter. The interrupt is reset and if execution in the foreground exceeded the proper interval, an error indicator is incremented. If the error occurred, immediate return is made to that point of interruption for processing continuation. The previous cycle will be completed and the new cycle will be lost. Normally, this error condition would not occur and the simulation would be called. However, if console sense switch (CSW) 1 is set, the simulation executive would not be called. Return to the Background is made at the conclusion of the Foreground. See Figure 50 for Foreground Task Flow.

The foreground task is primarily intended to provide the timing for the simulation of the ACV-LC. This timing is necessary to provide the proper time intervals for the mathematical modelling techniques of real-time simulation. Although the software description to this point has only been concerned with the timing structure and overhead, the primary objective of simulation is accomplished in the Foreground Task. The mathematical modelling of the ACV-LC is controlled by the simulation executive ACVSIM. Consequently, operation in real-time necessitates a call to ACVSIM at every time period.

The Counter 3 Zero Interrupt occurs at regular intervals of 50 msec (20Hz). This interrupt activates the Foreground Task and is handled by the routine EXECACV1. The timing of the foreground occurrence can be altered by changing the variables RUNFREQ and EXCOUNTI. RUNFREQ is the integer value of the frequency (20 Hz). EXCOUNTI is the integer value -8000/RUNFREQ (-400). In order that real-time be realized, RUNFREQ should divide into 8000 an integer number of times exactly. EXCOUNTI is used to reset the Counter 3 Interrupt. RUNFREQ and EXCOUNTI are the only variables used by the program to determine the foreground timing. However, the sampling period for real-time simulation should correspond to the interrupt period (1/RUNFREQ). DELTAT is a floating point variable and is the sampling period used in the simulation mathematical model. The value equals .05 sec.

ACVSIM

The simulation executive (ACVSIM) controls the execution of the mathematical model. (See Figure 51). Upon entry, the CRT display flag (CRTFLAG) and the VGW execution flag (VGGOFLAG) are reset. Simulation moding switches are tested to determine the path of execution:

SSW 1	initial conditions (IC) or reset
2	hold or freeze
3	operate or run

EXECACV1

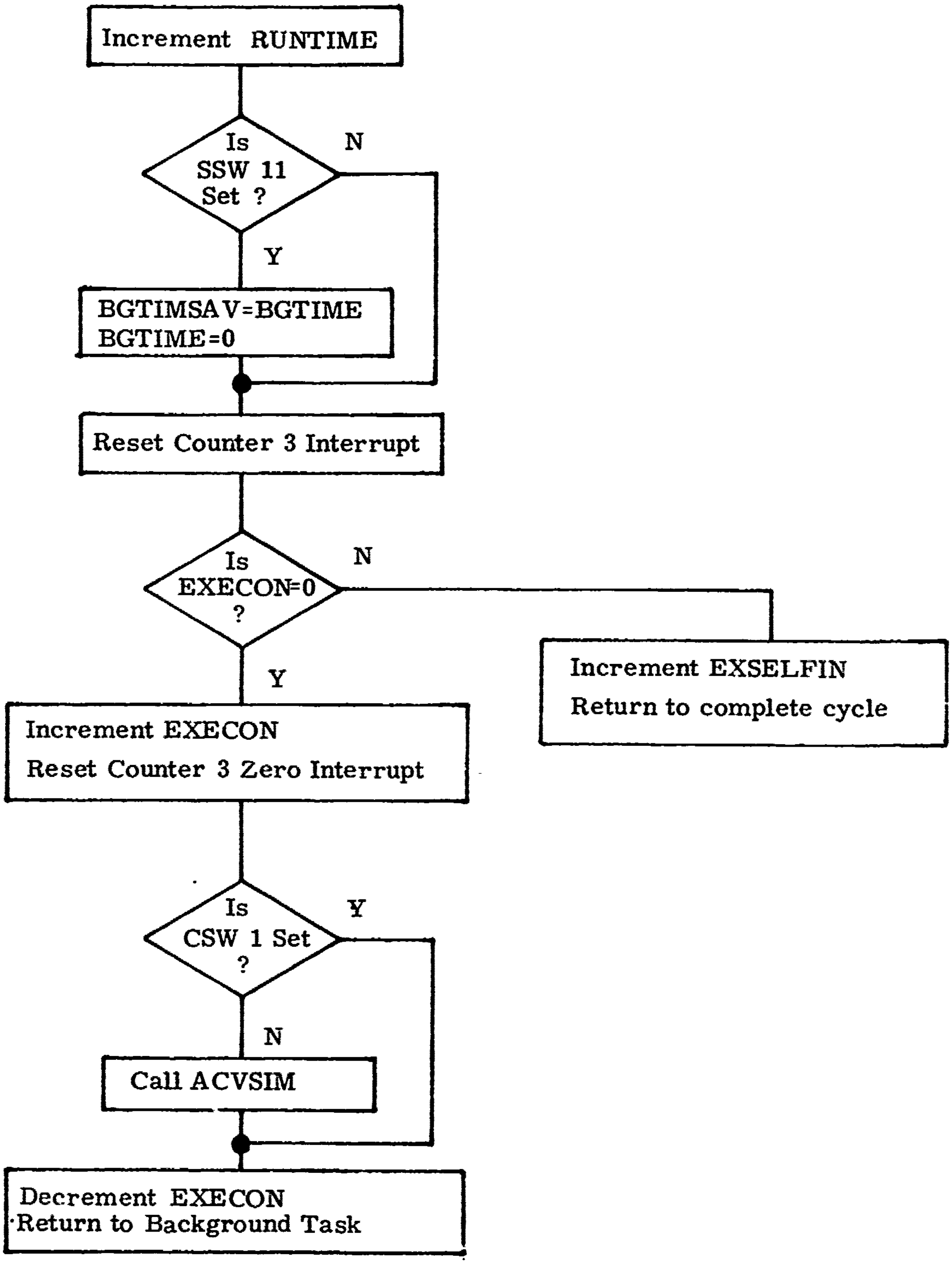


Figure 50. Foreground Task Flow

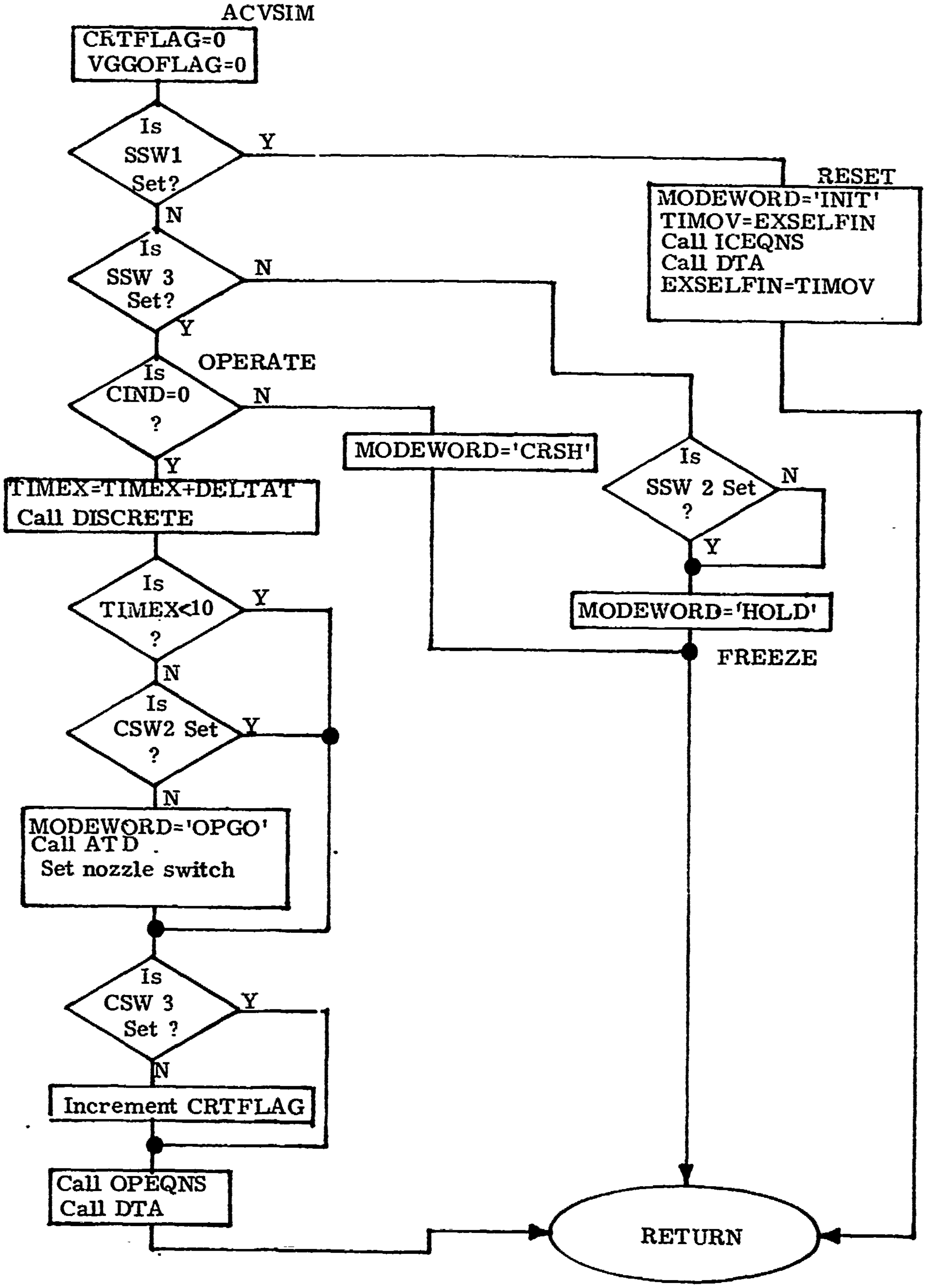


Figure 51. Simulation Executive Flow-ACVSIM

Any combination of mcre than 1 mode switch on or all off results in the hold mode, which is a donothing mode.

The reset mode causes initializaticn of the ACV-LC to a stable condition at zero speed and specified position. (See Figure 52 for flow) The time overflow indicator cannot be incremented and the cockpit displays are updated to correspond to the initial conditions.

The operate mode is the real-time simulation of the ACV-LC. The vehicle is operational and maneuverable in this mode. Each cycle in the operate mode first tests a flag (CIND) for a collision with the mother ship. If a collision has occurred, the hold mode is entered. Second, execution time (time since entering operate from IC) is updated. Third, the input discrete word is read to determine cockpit switch status. Fourth, if execution time is less than 10 seconds or if console sense switch (CSW) 2 is set, the pilot commands are bypassed. If execution time is greater than 10 seconds or CSW 2 is reset (the normal case), then the pilot commands from the cockpit are read into the simulation for use in the model equations. Fifth, if CSW 3 is not set, CRTFLAG is incremented to enable the CRT display. Sixth, the mathematical model equations (OPEQNS, see Figure 53) are called for execution, and then the displays in the cockpit are updated.

The variable MODEWORD is set to a value of characters corresponding to the mode of operation:

'INIT'	IC, reset
'HOLD'	hold, freeze
'CRSH'	hold, mother ship collision
'OP'	operate without pilot commands
'OPGO'	operate normally

This set of characters when printed is an immediate indicator of system status.

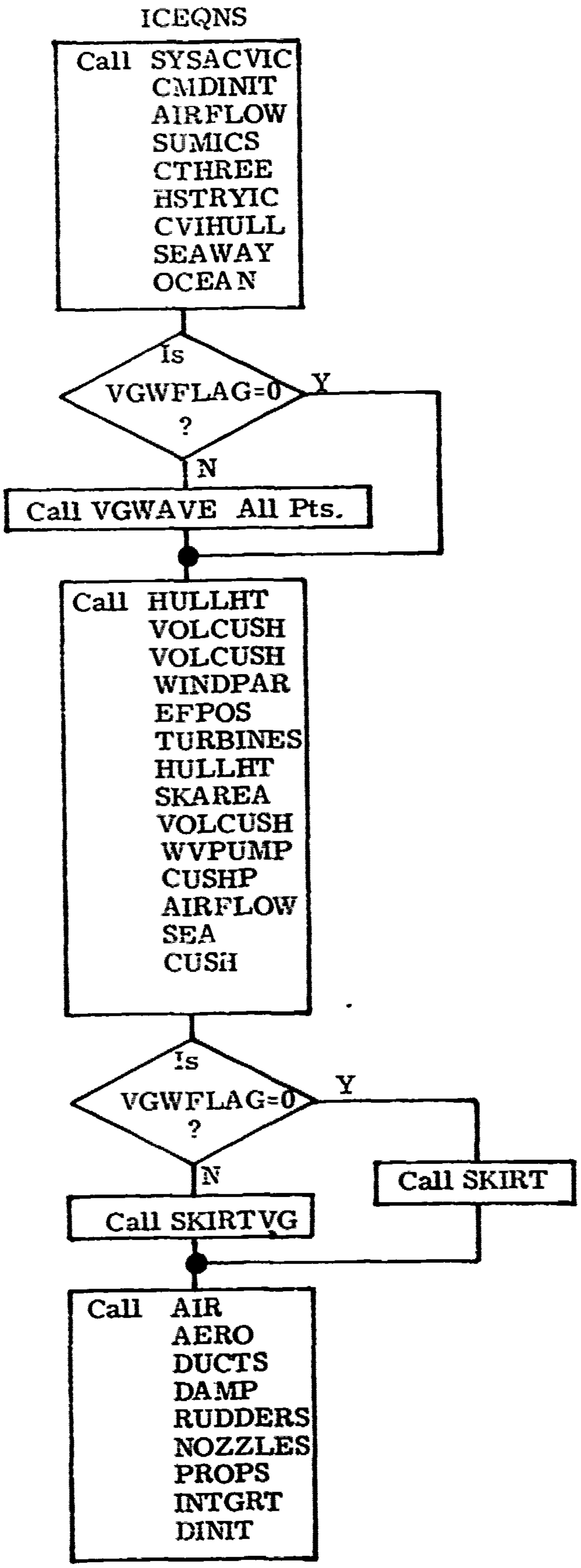


Figure 52. Initialization Equations Flow- ICEQNS

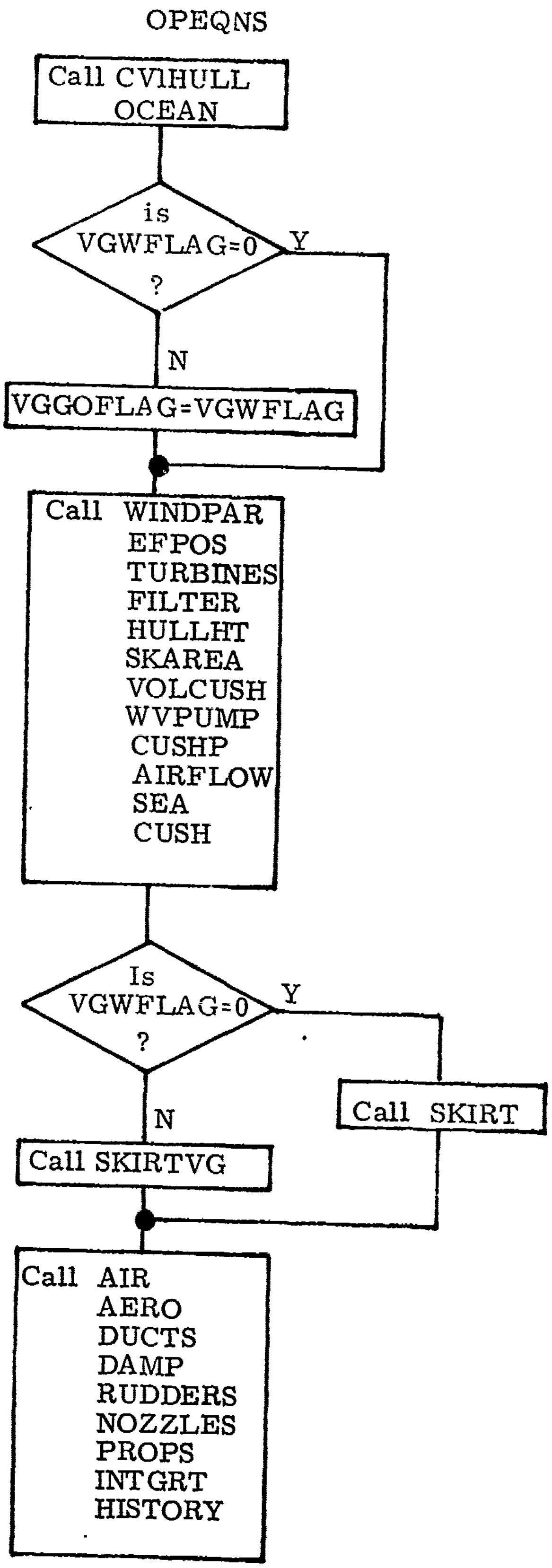


Figure 53. Operational Equations Flow - OPEQNS

?RAP HANDLING

Traps cause an interruption in program flow. Returning to flow continuation is optional and depends upon its implementation (and operation, in some cases). The Foreground has already been discussed, but is a trap (Counter 3 Zero) interrupting the Background and does return for continuation. Other traps are activated for a simple task and return immediately for continuation:

Counter 3

Control Panel

The Power Up interrupt causes an unconditional branch to the entry INITSTRT. The Power Down interrupt causes an infinite donothing loop. Restart is not really a trap, but simply a means of initializing and/or restarting the simulation program. When the operator pushes SYSTEM or CPU RESET the computer initializes itself to memory location X'26'. If the operator pushes RUN an unconditional branch to entry INITSTRT is caused.

CAL1 is a software caused trap which is used to cue I/O procedures. CAL1 is only used for those procedures involving the reading of the V.GW table. This trap is activated to perform an I/O function and then returns to the program for continuation.

All abnormal errors are hardware cued and are caused by software and/or hardware faults. These faults are handied by traps. If an abnormal trap occurs, the CAL4 trap is activated (see Figure 54) to record the error, turn on the audible alarm, and then places the simulation program into a WAIT state for operator intervention. The fault traps which are handled in this manner are:

0	Memory.parity
1	Non-allowed operation
2	Unimplemented instruction
3	Push down stack
4	Fixed point arithmetic overflow
5	Floating point fault
6	Decimal arithmetic fault
7	Watch Dog timer runout

The error number is recorded in location C4ERR# and the address of the error is recorded in C4ERRLOC. If the Memory Parity trap occurred, the memory location which has incorrect parity is recorded in C4ERR MEM.

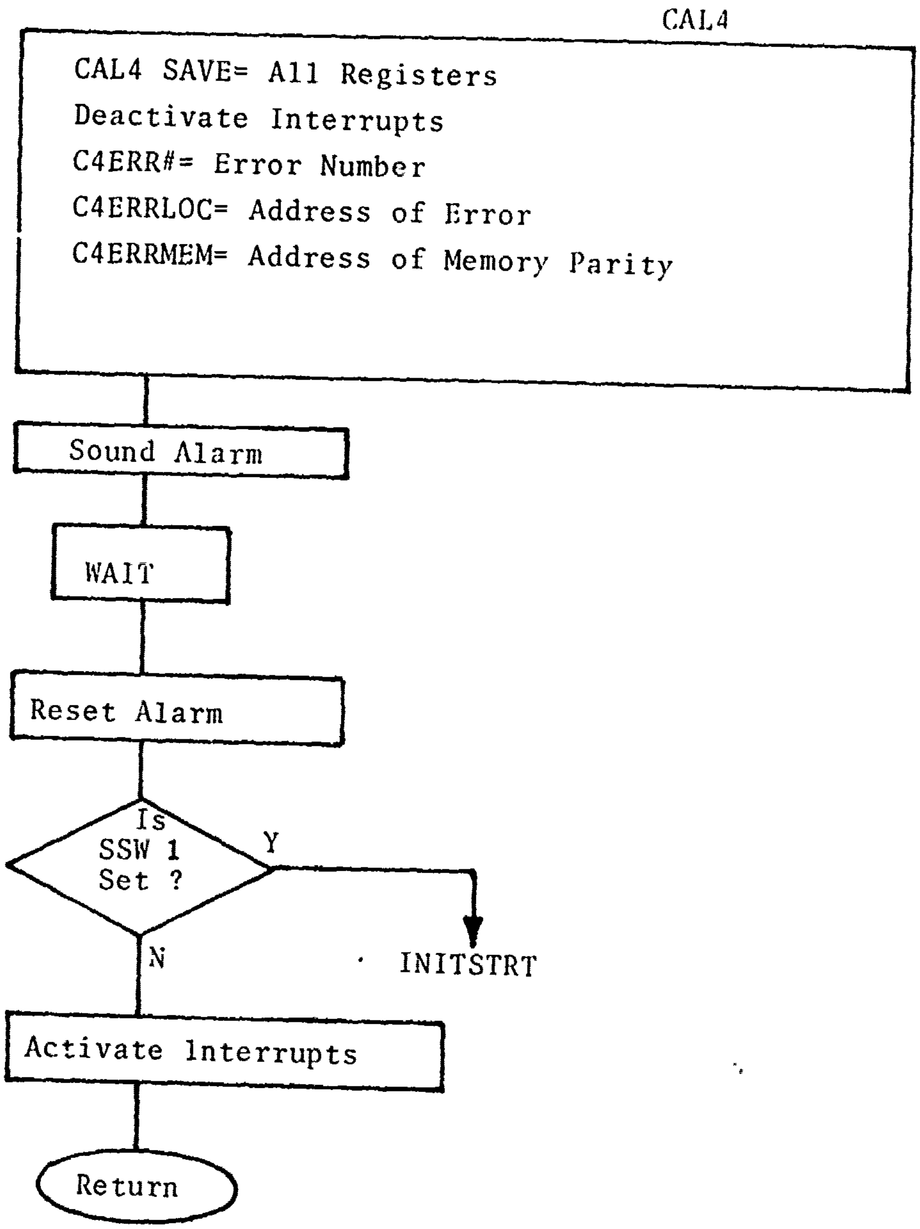


Figure 54 . CAL4 Error Processing Flow

COMMON STORAGE

The DATAPOOL acts as common storage for the complete simulation program. Most modules interface with the DATAPOOL in order to facilitate transfer of data from one module to another. This centralized location of data is also for the operator's convenience and is intended to minimize external definitions of variables within program modules.

The DATAPOOL is formatted for readability in the following way:

Forces and Moments
Summation Forces, Moments
State Vectors
Transformation Matrix
State for Line Drawing
Integration Increments (Not Used)
Change in Inertial Position
Constants about CG
Environmental Wind Factors
Special System Parameters
Equations of Motion
Propellers
Thrust Nozzles
Rudders
Aerodynamics
Propeller Ducts
Skirt
Manifolds
Compartment Volumes
Wave Pumping
Cushions
Gap Areas
Gas Turbines and Power Shafts
Seaway
VGW
Wave Heights
Hull Point Locations
Height of Hull above Water
Skirt Clearances
Input Discretes
Analog Inputs
Analog Outputs
I/O Parameter List
Typewriter Status Messages

MODULARITY AND TIMING

As has already been stated, the simulation program consists of a modular breakdown by function. That is, each module does a specified function or task within the system. Most modules are executed as subroutines with calls to them and returned from them. The structure has already been shown, so that it is obvious that EXECACV and READVGW are entered by unconditional branches. The major breakdown of the Background Task has transfers from major modules by unconditional branches.

All subroutines link by using Register 15 except those whose names begin with a digit (See Table 10). Subroutines which have names beginning with a digit link by Register 6 and are library functions. The I/O library functions link by Register 15.

The digit 6 preceding the name signifies that the module was taken from the Sigma 7 FORTRAN library. The digit 7 preceding the name signifies that a special numerical technique was developed for rapid execution as a library function. The digit 1 preceding the name signifies that the module was specially developed to solve a particular numerical task within the mathematical model and are not library functions but use Register 6 as the linkage. Secondary entry points within the major modules of Table 10 follow the same rules. See Table 11 for the list.

The timing of most real-time subroutines has been indicated in the tables. These numbers were obtained under the monitor and are not necessarily accurate as the monitor does not always respond at the proper time. Nevertheless, these timing numbers are good indicators of relative timing as they were obtained by executing each subroutine 1000 times. The timing for the library functions was obtained by either taking the timing indicated from the FORTRAN library listings or by counting instructions.

Individual timing of each subroutine is not important by itself, but the proper sum in the operate mode is. From the table, the timing of OPEQNS is 29.372 m sec. This number represents the system off-line without FILTER and without the real-time simulation overhead of mode determination, reading A/D's and discretes, and outputting D/A's. From actual dwell time of the real-time system, 29.2 msec represents the time spent executing equations in the deep ocean (29.2/50= 58%) and 40.5 msec represents the time spent in the ocean shoaling region (40.5/50= 81%).

Major Simulation Modules

*

Module	Description	Timing(MS)	BG	IC	OP	LIB	CALLER	VGW
1KINTRP	Retrieve kernel value for VGW		X	X		X	VGWAVE	
1SIMEQ	Limited 6 x 6 matrix solution			X	X	X	CUSHP	
1WVNBR	Wave number and phase	.143		X	X	X	OCEAN	
6ATAN	Arctangent function	.130		X	X	X	WINDPAR	
6COS	Cosine function	.089		X	X	X	OCEAN	
6SQRT	Square root function	.072		X	X	X	WINDPAR	
7ASQRT	Special square root function			X	X	X	CUSHP	
7DASQRT	Derivative of 7ASQRT			X	X	X	CUSHP	
7SGNSQR	Limited signed square root function	.029		X	X	X		
7SINCOS	Limited sine and cosine functions	.082		X	X	X		
7SYNCRO	Limited sine and cosine functions	.050		X	X	X	DTA	
ACVSIM	Equations executive			X	X		EXECACV	
AERO	Aerodynamic forces and moments	.205		X	X			
AIR	Air momentum forces and moments	.061		X	X			
AIRFLOW	Air flow rates	.381		X	X			
ATD	Read and format analog voltages						ACVSIM	
BACKGRND	Dwell time determination		X				PRINT	
BCDATA	EBCDIC to FX, FS, X		X					
BCDCOMMA	Comma delineated fields		X					
BCDPARAM	Search for parameter address		X					
BCDSTORE	Reads 'Param' and 'Memory'		X					
CAL1HAN	Call handler						EXECACV	X
CARDIO	Process card image		X				CARDS	
CARDREAD	Read card image (M:READ)		X				CARDIO	

TABLE 10. Major Simulation Modules

Module	Description	Timing(MS)	BG	IC	OP	LIB	CALLER	VGW
CARDS	Call CARDIO		X				TYPEMSG	
CMDINIT	Initialize pilot commands	.089		X			ICEQNS	
CTHREE	Coordinate transformation	.336		X	X			
CUSH	Cushion pressure forces & moments	.133		X	X			
CUSHP	Compartment pressures	3.860		X	X			
CVIHULL	Transform hull pt. locations to inertial	1.178		X	X			
DAMP	Aerodynamic moment	.011		X	X			
DATAPOOL	Common storage							
DECIT	Integer to EBCDIC		X			X		
DISCRETE	Input discrete switches				X		ACVSIM	
DLOOP	CRT Graphic display		X				DLOOPER	
DLOOPER	Data conversion and call DLOOP		X				BACKGRND	
DTA	Format and write analog voltages			X	X		ACVSIM	
DUCTS	Propeller duct forces and moments	.493		X	X			
EFPOS	Pilot controls to effector commands	.300			X		OPEQNS	
EXECACV	Process interrupts and traps						READVGW	
FILTER	First order lag on VGW heights			X	X			
FIXPRNT	F-Format to EBCDIC		X			X		
FLTDEC	Floating point to decimal and exponent		X			X		
FLTPRNT	E-Format to EBCDIC		X			X		
FORCES	Summation forces & moments	.618		X	X		INTGRT	
GRAVITY	Gravity forces and moments	.059		X	X		INTGRT	
HEXIT	Hexidecimal to EBCDIC		X			X		
HISTORY	ACV Trajectory history	3.004			X		OPEQNS	
HSTRYIC	History initialization	1.291		X			ICEQNS	

TABLE 10. Major Simulation Modules (Cont'd)

Module	Description	Timing(MS)	BG	IC	OP	LIB	CALLER	VGW
HULLHT	Average height of hull	1.059		X	X			
ICEQNS	Equations for initialization	30.684		X			ACVSIM	
INTGRT	Vehicle state integrations	2.536		X	X			
MOTION	Equations of motion	.944		X	X		INTGRT	
NOZZLES	Thrust nozzle forces & moments	.123		X	X			
OCEAN	Dynamic seaway generation	12.730		X	X			
OPEQNS	Real time equation‘	29.372			X		ACVSIM	
PRINT	Call SYSPRINT		X				CARDS	
PRNTLINE	Transfer 2 line to printer(M:Print)		X			X		
PRNTPAGE	Printer to top of page (M: Devide)		X			X		
PRNTTIME	Provide date & time in EBCDIC (M:Time)		X			X		
PROPS	Propeller forces and moments	.192		X	X			
PRTDATA	Print DATAPOOL contents		X				PRINT	
READMTØ	Transfer 1 record from 9T-MT(M:Read)					X	CAL1HAN	X
READVGW	Read VGW kernel table from 9T-MT							X
RUDDERS	Rudder forces and moments	.159		X	X			
SEA	Seaway & VGW forces and moments	.424		X	X			
SEAWAY	Seaway initialization	.223		X			ICEQNS	
SKAREA	Skirt escape area	.486		X	X			
SKIRT	Skirt & spray drag forces & Moments	.056		X	X			
SKIRTVG	Skirt drag forces & moments for VGW	.062		X	X			
SUMICS	Integrations summing	.184		X	X			
SYSACVIC	Equations initialization	.195		X			ICEQNS	
SYSPRINT	Print ACV State		X				PRINT	

TABLE 10. Major Simulation Modules (Cont'd)

Module	Description	Timing (MS)	BG	IC	OP	LIB	CALLER	VGW
TURBINES	Turbine & Output Shaft Characteristics	1.216		X	X			
TYPE	Process typewriter EBCDIC		X				BACKGRND	
TYPEIN	Read 1 Line from typewriter		X					
TYPEINØ	Read 1 line from typewriter immediate					X		X
TYPEMSG	Type messages to operator		X				TYPE	
TYPENL	Type new line		X			X		X
TYPEOTØ	Type 1 line immediate					X		X
TYPEOUT	Type 1 line		X					
VGWADDR	VGW kernel table addresses		X	X			1KINTRP	X
VGWAVE	Vehicle generated wave heights		X	X				
VGWTABL	VGW kernel table		X	X				
VOLCUSH	Cushion compartment volumes	.279		X	X			
WINDPAR	Environmental wind factors	.261		X	X			
WVPUMP	Wave pumping air flows	.219		X	X			

* BG	Background execution
IC	Foreground execution-initial calculations
OP	Foreground execution-real-time calculations
LIB	Library routines
CALLER	Module called from
VGW	Kernel table read from 9T-MT

TABLE 10. Major Simulation Modules (Cont'd)

Secondary Entry Points

Entry	Module	Description	Timing(MS)	BG	IC	OP	LIB	CALLER	VGW *
7SCDEG	7SINCOS	Unlimited sine & cosine, degrees	.127		X	X	X		
7SCRAD	7SINCOS	Unlimited sine & cosine, radians	.128		X	X	X		
CAL11	CAL1HAN	M:Rew, M:Device, M:Read, M: Write						EXECACV	X
CAL12	CAL1HAN	M: Print, M: Merc						EXECACV	X
CAL18	CAL1HAN	M: Time						EXECACV	X
CARDS OFF	CARDS	CARD IO exit		X					
DINIT	DLOOP	Initialize CRT display			X			ICEQNS	
EXITCAL1	EXECACV	Call exit						CAL1HAN	X
INITSTRT	EXECACV	Initiate executive						READVGW	X
READVGWI	READVGW	Read VGW table immediate							X
RDVGWBPM	READVGW	Read VGW table under monitor							X
TRAPSET	EXECACV	Set trap locations						READVGW	X
TYPEWAIT	TYPE	Continue typing functions		X					

* Entry — Secondary entry point

Module — Within a major module

TABLE 11. Secondary Entry Points

SECTION V

OPERATIONS

SYSTEM DESCRIPTION

The ACV Simulation can logically be described in three sections: the real-time equations, the background I/O, and the traps.

The largest section of core is taken by the real-time ACV equations. The timing for these equations is provided by the Counter 3 Zero Interrupt. At present the Count 3 Zero Interrupt is programmed to run at 20 Hz. The processor for the Counter 3 Zero Interrupt is contained in the program EXECACV which contains all the ACV interrupt processors. The Counter 3 Zero Interrupt calls the simulation program ACVSIM. ACVSIM drives all the ACV math model programs.

The timing of the ACV simulation can be changed from 20 Hz by altering the locations RUNFREQ, EXCOUNTI, and DELTAT. RUNFREQ is the integer value of the frequency which is 20 at present. EXCOUNTI is -8000/RUNFREQ and is -400 at present. EXCOUNTI is used to reset the Counter 3 Interrupt. DELTAT is a floating point single value. It is the integration time constant; at present set to .05 (1/20). In order not to lose time in the real time simulation, RUNFREQ should divide into 8000 an integer number of times.

The ACV system has been programmed to allow the real time equations to interrupt themselves. If this occures, the equations will not be restarted during the new cycle. The last cycle will be completed from the point of interruption and the rest of the cycle will be spent in the background loop. This event will be recorded by the Counter 3 Zero Interrupt. The interrupt will increment the location EXSELFIN each time it interrupts the real time programs.

The background I/O programs are continually being cycled through whenever the real time equations are not being executed. The background programs are made up of six subsections. These subsections are; the background timer, the CRT driver, the typewriter input processor, the typewriter error message processor, the card reader processor, and the line printer processor. These sections contain all the I/O for the respective peripheral equipment.

The background timer is requested by the operator by setting system sense switch 11(SSW11). The background timer is a counter that keeps track of the length of time that background is in operation. The value of the counter is saved and reset each cycle by the real time interrupt. Subtracting the background time from the time of a complete ACV cycle will provide the time that the real time equations are using. The timer is incremented in units of 12 microseconds. It is saved in location BGTIMSAV. As long as the background timer is being calculated (i.e., SSW11 is set), the peripheral equipment and the CRT cannot be used. The background timing program occupies the entire time the simulation is in the background state.

The CRT driver will display visual cues to the cockpit operator during the ACV simulation. It will not be executed if console sense switch 3(CSW3) is put on.

The typewriter input processor reads and processes operator requests to the typewriter. Requests can be made to the typewriter program as long as the typewriter input light is lit. This light is set on during system initialization. It can be put on or off alternately by triggering the console interrupt. When type requests are not being made, this light should be put off.

The typewriter error message processor will print error messages on the typewriter. These messages cannot be typed if the typewriter input light is lit.

The card reader processor reads cards. Card reading is requested by the operator by setting SSW8.

The line printer processor prints system parameters onto the line printer. This is requested by setting SSW 9.

The third section of the ACV simulation is the error trap processor. When an error trap occurs, the trap in turn calls the trap CAL 4. CAL 4 will save all the register values plus some diagnostic information. Seven types of errors can occur. (Memory parity is not really a trap but is treated like one in the error routine).

Error #	Error
0	Memory parity
1	Non-allowed operation
2	Unimplemented Instruction
3	Push down stack fault
4	Fixed point arithmetic overflow
5	Floating point fault
6	Decimal arithmetic fault
7	Watch dog timer

Three words are saved, along with the data registers, when one of these errors occurs. C4 ERR# contains the number of the error that occurred in the right most hexadecimal character. C4 ERRLOC is the PSW1 of the PSD of the trap that occurred. This word contains the memory location that caused the error (except memory parity error) and the condition code bits that further define the error. C4 ERRMEM has the memory module fault bits in bits 24-31. If an error occurs the console alarm is triggered. The computer will go into the WAIT state. There are three ways to recover from the WAIT condition. The program will continue from the location that caused the error, if the operator sets console sense switch 1 (CSW1) off and puts the computer in the IDLE and then RUN modes. The system will be reinitialized if the operator sets CSW1 on and puts the computer in the IDLE and RUN state. An alternate method of reinitializing the system can be done by putting the computer in IDLE, pressing CPU RESET, and putting the computer in the RUN mode. The last method will reinitialize the ACV system at any time during the ACV Simulation.

Provisions have been made for processing power on and power down interrupts. The power on interrupt will restart the system and the power down interrupt will force itself into an infinite loop.

INITIAL LOAD

The ACV Simulation System is contained on magnetic tape. Boot the tape into the computer with console sense switch 1 (CSW1) on and all the system sense switches (SSW) off. CSW1 inhibits the real time interrupt from calling the program ACVSIM. CSW 1 is in effect an alternate freeze mode.

The typewriter will print:

TYPE IN TIME (HH:MM) AND DATE (MON DD,'YY)

and the typewriter input light will be put on. The time and date should be typed in. The simulation will run if these values are not entered, however.

The format of the time input is:

TIME HH:MM

HH is the time in hours (0-23) and MM is the time in minutes (0-59). The spacing is free format except there must be one and only one character between HH and MM. (; is optional. Any character will do, including a space).

The format of the date input is:

DATE MON DD,'YY

MON is the three character abbreviation of the current month (e.g., JAN, FEB, MAR, etc.) DD is the day of the month (01 to 31); 'YY is the last two digits of the year (e.g., '75). The spacing of the data input is free format but the format of MON DD,'YY is fixed. There must be one and only one space between the month and the day.

Unless other type-in requests are made, the typewriter input light should be put off by activating the console interrupt. Other type-in requests will be discussed below.

The ACV simulation is controlled by the system sense switches 1, 2, and 3.

SIMULATION OPERATION

The ACV simulation is controlled by the system sense switches 1, 2, and 3.

SSW1 is on for RESET mode

SSW2 is on for FREEZE mode

SSW3 is on for OPERATE mode

One and only one switch should be on at any one time. Any other combination (two or three of the SSWs on or all of them off) of switches 1, 2, and 3 will put the simulation in the FREEZE mode.

When CSW1 is on (as given in the load procedure above) none of the system sense switches are tested. Their positions at this time are immaterial. To start the ACV simulation, put the system into RESET by putting SSW1 on and SSW3 off. Put CSW1 off. CSW2 and CSW3 should also be off at this time. The computer is now continually resetting the simulation parameters.

To put the ACV simulation in OPERATE set SSW1 off and SSW3 on. The FREEZE mode will be entered momentarily until the proper moding is achieved.

To freeze the simulation SSW2 should be set to on. The state of the other moding switches is not important. FREEZE mode is in effect a system no-op. No simulation I/O or system equations will be executed. I/O on peripheral equipment such as the typewriter can be done, however.

CSW1, CSW2, and CSW3 are used to inhibit different parts of the ACV system. CSW1 inhibits the entire simulation (the call to ACVSIM and the CRT driver). CSW2 inhibits the commands from the cockpit to the simulation. CSW3 inhibits the call to the CRT driver. CSW2 and CSW3 are only effective in OPERATE mode because the cockpit parameters and the CRT driver (DLOOP) are not referred to in RESET or FREEZE modes. The cockpit inputs and CRT outputs will remain at their last values until RESET is entered. They will be re-initialized at this time. CSW1, CSW2, and CSW3 do not effect the operation of the typewriter, the card reader, or the line printer.

The status of the ACV simulation can be monitored periodically. Momentarily setting SSW9 prints various simulation parameters on the line printer. IF SSW9 is left on, the data will be printed continuously. A description of the printed data can be found in ACV System Printout Description. Printing can be done in any simulation mode but the printed data would be more useful if printing is done in the FREEZE mode only. Printing the state variables cannot be done if the background timer is in operation (SSW11 is on) or if card reading is taking place.

The ACV system has the capability of altering or printing the contents of any location in memory. Locations can be altered with either the card reader or the typewriter. Locations can be printed on the typewriter. The capability to print random memory location is an additional feature separate from the line printer status printout. The typewriter and card reader I/O are inhibited if SSW11 is on (i.e., the background timer).

CARD MODIFICATION

To alter locations from the card reader, SSW8 should be set. Cards will be read and printed on the line printer. No cards will be processed until the first keyword card is read. This card has the word:

ACVINIT

on it, starting in column one. Cards will continue to be read, regardless of the state of SSW8, until the second keyword card is read. This card has the word:

ENDACVIN

on it, starting in column one. All the cards from ACVINIT to ENDACVIN will be processed for valid memory alterations.

There are two ways to change memory locations using the card reader. The first is by using the keyword MEMORY. The second is by using the keyword PARAM. Every word in memory can be changed with the MEMORY keyword. The PARAM card uses pre-assigned parameter names to access memory. Only a small number of often used locations can be accessed. PARAM cards do not have to be changed after reassemblies.

The formats of the MEMORY card and the PARAM card are:

MEMORY ADDRESS, DATA, DATA,...

PARAM PARAMETER, DATA, DATA,...

In both cases the keyword must start in column one. After the keyword the spacing is free format. ADDRESS is a numeric value and is assumed to be an absolute memory address. PARAMETER is an alphanumeric word internally associated with a specific memory address. The address, whether absolute or parametric, will be the starting address in which to store the data that follows the address. The DATA values are stored in consecutive memory locations, starting with the first address. The format of the DATA values includes a value refered to as a multiple. It is used to repeatedly store a single DATA value in a number of consecutive locations. That is, if a DATA value has a multiple of 10, 10 values of DATA will be stored into locations ADDRESS to ADDRESS + 9 (or PARAMETER to PARAMETER + 9).

ADDRESS has the same format as the DATA words. However, any multiple that is provided with the ADDRESS value is ignored. PARAMETER's are preassigned alphanumeric words. A list of those words that can be used as PARAMETER's is given in Parameter Values for Initialization. The format of the DATA value is given below.

If the MEMORY card or the PARAM card ends with a comma, data can be continued onto the next card. Data on a continuation card can start on any column. Continuation cards can also have continuation cards. A card that ends with a comma does not necessarily have to have a continuation card. A valid keyword in column one will terminate a continuation string. Only DATA words can be on continuation cards. The ADDRESS or PARAMETER word followed by a comma must be on the first card.

If any part of a card cannot be translated, an error message will be printed on the line printer underneath the card's printed image. The operator will have to localize the error on the card. An error does not cancel the entire card, however. If a valid address and a modification value can be determined by the card processor before the error is found, that location modification will be performed. Data on the card from the error to the end of the card will be ignored and any continuation cards will be ignored.

DATA words have one of the following formats:

MMM	D'±DDDDDD'	decimal integer
MMM	D'±.DDDDDD	decimal fraction
MMM	F'D.DDDDDE±DD'	floating point
MMM	X'HHHHHH'	hexadecimal

In all cases MMM is the multiple value mentioned previously. These characters must be decimal digits and can be any number of characters as long as the word does not overflow a 32 bit word magnitude. The multiple cannot have a sign and is always positive. There must be at least one blank after the multiple.

The decimal integer and fraction formats (D) represent either a 32 bit decimal integer scaled B31 or a 32 bit decimal fraction scaled B0. The following rules apply.

1. Data must start with D'
2. The sigh (+ or -) is optional. Plus is assumed.
3. The first character can be a decimal character (0-9) followed by a single quote or a decimal point (.).
4. The remaining characters must be decimal characters (0-9) followed by a single quote (').

The decimal point cannot be preceded by any decimal characters. Too many characters in an integer will cause an arithmetic overflow. Extra digits in a fraction will be rounded.

The floating point format (F) represents a Sigma 7 single floating point number. These rules apply.

1. Data must start with F'.
2. The sign of the mantissa (+ or -) is optional. Plus is assumed.
3. The first character must be a decimal digit (0-9).
4. The second character can be a decimal point or a decimal digit. If the decimal point is left out, it will be assumed to be after the first digit (3 above).
5. The mantissa can be continued to any number of digits. It will be rounded.

6. The exponent (E±DD) is a power of 10. It is optional and if left out, E0 is assumed. Its sign is optional. It can have either one or two decimal digits.

7. The floating format must end with a single quote (').

The hexadecimal format (X) represents a 32 bit hexadecimal value. The rules for the hex format are.

1. Data must start with X'.

2. Any number of hex characters can be used.

3. Data must end with a single quote (').

The data is stored right justified with left characters zeroed. If the data word is more than eight characters long, the right most eight characters will be used. Hexadecimal data words have no sign.

In all DATA formats there can be no embedded blanks. DATA values cannot be continued from one card to another. Some examples of acceptable and non-acceptable DATA words are given below.

Acceptable Data

Decimal Integer

DATA	Meta-Symbol Equivalent
D''	0
D'2'	2
D'+48'	48
D'-3976'	-3976

Decimal Fraction

DATA	Meta-Symbol Equivalent
D''	FX'0B0'
D'.5'	FX'.5B0'
D'+.623'	FX'.623B0'
D'-.9289'	FX'-.9289B0'
D'-.62314383678901543'	FX'-.623143837B0'

Floating Point

DATA	Meta-Symbol Equivalent
F''	FS'0'
F'5'	FS'5'
F'+5'	FS'5'
F'-6.2'	FS'-6.2'
F'-628E2'	FS'-628'orFS'-6.28E2'
F'-7.45E-10'	FS'-7.45E-10'
F'4.666666666666'	FS'4.66666667'
F'+1800E3'	FS'1800' or FS'1.800E3'
F'2.78E+28'	FS'2.78E+28'

Hexadecimal

DATA	Meta-Symbol Equivalent
X''	X'0'
X'ABC'	X'ABC'

DATA	Meta-Symbol Equivalent
X'ABCDE123456789'	X'23456789'
X'ABCDDCBA'	X'ABCDDCBA'

Unacceptable Data

Decimal Fraction

DATA	Reason For Error
D'6.'	decimal point in wrong place
D'.999999999999'	data overflow after rounding

Floating Point

DATA	Reason For Error
F'65.7'	decimal point in wrong place
F'7.82E'	no exponent characters
F'.35E2'	decimal point in wrong place
F'5.32E-106'	too many characters in exponent
F'9.999999999999E-1'	data overflow after rounding

None of the examples above were given multiples but there is no reason why they could not have had them. The following lines are examples of MEMORY and PARAM cards having DATA values with multiples attached.

```
MEMORY X'2A01',F'2.0',10 D'48'
MEMORY X'1BCD',28 X'1234',D'14',
F'6.E20', 4284 D'-64'
PARAM U0,F'450E2'
PARAM WIND,F'4.5E1'
PARAM MATRIX0,9 F'0'
PARAM MATRIX1,F'1',3 F'0', F'1',
3 F'0', F'1'
```

TYPEWRITER MODIFICATION AND DISPLAY

The typewriter can be used to display and alter memory locations and to set or change the date and/or time. Type-ins can be done any time during the ACV simulation as long as the typewriter light is on. This light is alternately put on and off by activating the console interrupt.

The time and the date input were described above. Memory locations can be altered by a MEMORY or a PARAM type-in. The format of these type-ins are almost the same as the card MEMORY and PARAM inputs. The only difference is that the keywords MEMORY and PARAM do not have to start on column 1. Continuation lines can be used with the PARAM and MEMORY type-ins.

The contents of memory can be typed onto the typewriter by the type keyword DUMP. The formats of the DUMP type-ins are:

DUMP D DATA	decimal type-out
DUMP X DATA	hexadecimal type-out
DUMP E DATA	mantissa-exponent type-outs of single floating point values
DUMP F DATA	integer fraction type-out of single floating point values

The spacing of the type-ins are free format. As many spaces as desired (including none) can be typed between DUMP and the format characters (D,X, E or F) or between the format character and the DATA value. DATA represents a multiple-value pair as described on the MEMORY and PARAM DATA words. In this case the multiple, if present, is the number of word locations to be typed out. The value of the DATA word is the starting address of the consecutive locations to be typed. DUMP type-ins cannot be continued on a new line. The DUMP type-in will type out pairs of values which represent the address and the contents of the address of the requested locations.

The four different output formats are as follows:

D The decimal equivalent of the word address contents. Leading zeros are suppressed and the value is left adjusted. The format is SDDDDD. S is the sign + or -. D's are decimal digits (0-9).

X The hexadecimal equivalent of the address contents. Leading zeros are typed. The format is HHHHHHHH. H's are hex digits. Eight characters are always typed.

E The mantissa - exponent equivalent of the word address contents. The format is SD.DDDDDESDD. S is the sign, + or -, of the mantissa or the exponent. D is a decimal character (0-9). E is the character 'E' and represents a power of 10.

F The integer fraction equivalent of the word address contents. This is for a floating point value. The format is SDDDDDDD. S is the sign, + or -. D is a decimal character (0-9) or a decimal point. The position of the decimal point will vary depending on the magnitude of the value. Leading zeros are suppressed (except the value zero) and the number is left adjusted. If the value is too small, zero will be typed. If it is too large, ±999999999 will be typed.

For all type-ins the following rules apply. This includes TIME, DATE, MEMORY, PARAM, and DUMP type-ins.

1. Type lines should end with a new line character (NL).
2. A typed line will be terminated after 85 characters and will be rejected by the I/O processor.
3. To cancel a line, type in an end of message character (EOM)
4. The horizontal tab character (HT or TAB) acts like a NL character.

Incorrect characters (except for NL, EOM, or HT characters) can be corrected by typing a ¢ (cent mark} for each character that was previously typed and is to be deleted (e.g., DUNP¢¢MP is equivalent to DUMP).

ACV SYSTEM PRINTOUT DESCRIPTION

Heading Line:

ACV-LC, Time of Day, Date, Current System Status

The current system status consists of a 4 character code with values:

INIT	Reset Mode
HOLD	Freeze Mode
OP	Operate Mode without Pilot (or < 10 sec)
OPGO	Operate Mode (Simulation Operable)
CRSH	ACV-LC Hit Mother Ship

In body axis coordinates, the forces and moments are shown that act on the vehicle broken into the various components. The total forces and moments are shown and through Newton's Second Law, the body axis accelerations are given. Initial velocities are printed and by integration the change in velocity and the current velocities are printed in body axis coordinates.

Performing a coordinate transformation yields velocities in earth-fixed coordinates. Integration yields change in position and summing with initial positions yields the current position of the vehicle. Angular position is displayed in both degrees and radians.

The effectors consist of the various physical devices that directly cause the vehicle to maneuver, pilot commands to these devices where applicable, and other conditions which are measurable, and environmental effects. Refer to Figure 55 for a sample printout.

The list is as follows:

STBD and PORT propeller speeds (NPS, NPP) (RPM)

STBD and PORT propeller pitch (BETAPS, BETAPP) (DEG)

Rudder Angle (PSIR) (DEG)

Thrust Nozzle Angle (PSIN) (DEG)

STBD and PORT propeller thrust (TPROPS, TPROPP)(LBS)

STBD and PORT propeller pitch commands (BETAPSC, BETAPPC)(DEG)

Rudder Command (PSIRC) (DEG)

Nozzle Command (PSINCA) (DEG)

Cushion Compartment Pressures (PCUSH 1, PCUSH 2, PCUSH 3, PCUSH4)
(LBS/FT^2)

Manifold Pressures (PMANS, PMANP) (LBS/FT^2)

Air Flow through STBD and PORT fans (QFANS, QFANP)
(FT^3/SEC)

Air Flow through STBD and PORT Nozzles (QNOZS, QNOZP)
(FT^3/SEC)

STBD and PORT fan speeds (NFANS, NFANP) (RPM)

Wind Velocity (WIND) (FT/SEC)

Wind Direction - Set (PSIWIND)(DEG)

Cushion Compartment Volume (VCUSH1, V CUSH2 , V CUSH 3, V C USH4)
(FT^3)

Air Flow due to wave pumping into Cushion Compartments
(QPUMP1, QPUMP2, QPUMP3, QPUMP4) (FT^3/SEC)

Skirt Gap Area under each Compartment (AREA 1, AREA 2, AREA 3, AREA 4) (FT^2)

STBD and PORT power turbine speed (NSS, NSP) (RPM)

STBD and PORT gas turbine speeds (NTS 1, NTS 2, NTS 3, NTP 1, NTP 2, NPT 3) (RPM)

STBD and PORT power turbine commands (NSCS, NSCP) (RPM)

STBD and PORT gas turbine commands (NTCS 1, NTCS 2, NTCS 3, NTCP 1, NTCP 2, NTCP 3) (RPM)

Offshore wave height (HEIGHT) (FT)

Offshore wave period (PERIOD) (SEC)

Ocean bottom slope (SLOPE)

Offshore-shoaling boundary (XSHOAL) (FT)

Offshore wave length (LAMBDA) (FT)

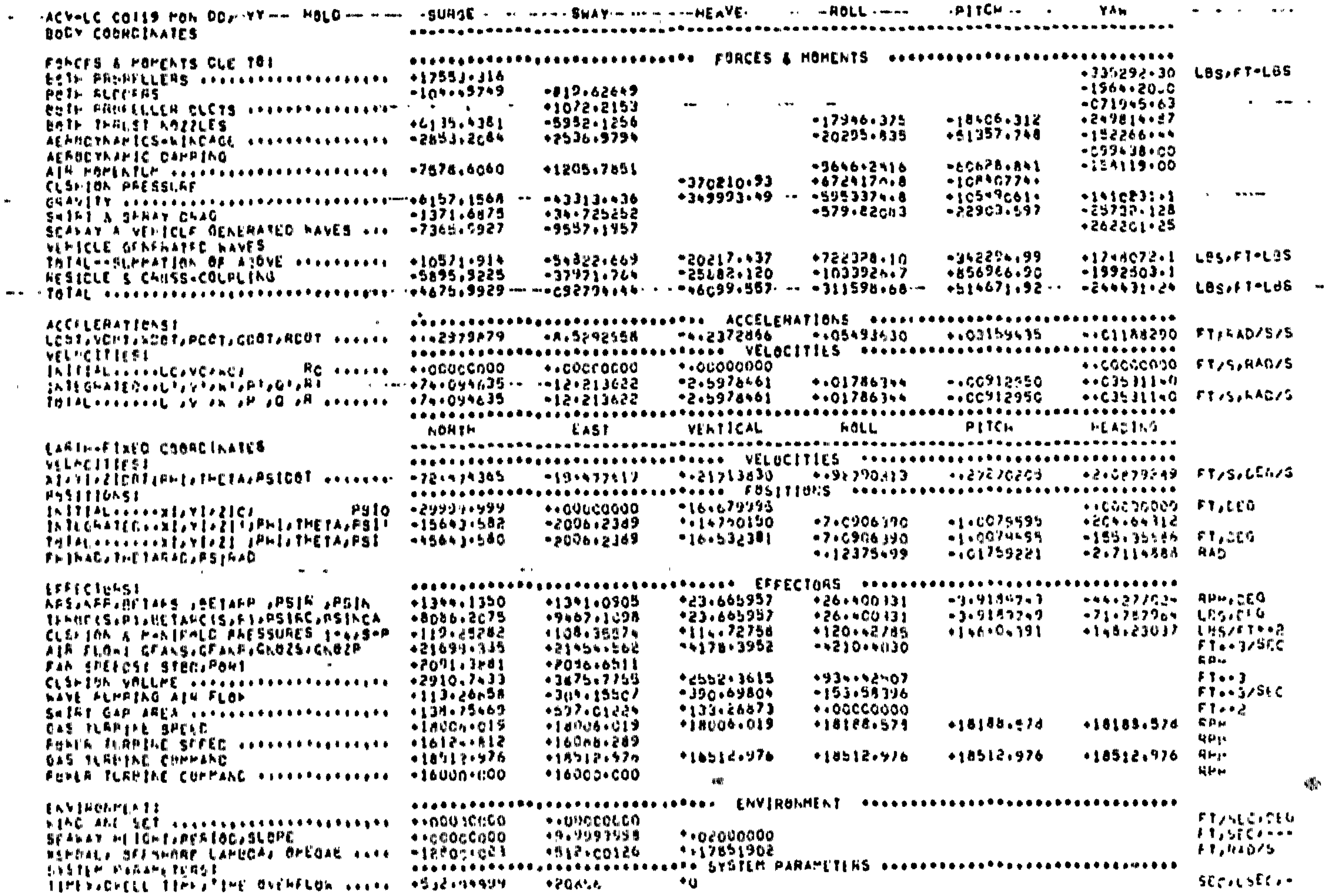
Wave encounter frequency (OMEGAE) (RAD/SEC)

System parameters are displayed to show specific status.

Execution Time (TIMEX) (SEC)

Dwell Time (μSEC)

Time Overflow Counter (EXSELFIN) (COUNTS)



```
ACV-LC C0119 MON DD/YY--- HOLD----    -SURGE-       -SWAY-        -HEAVE-       -ROLL-        -PITCH-       YAW
BODY COORDINATES                      ****************************************************************************

FORCES & MOMENTS DUE TO:              **************************** FORCES & MOMENTS *****************************
BOTH PROPELLERS .....................  +17553.316                                                           +335292.30  LBS/FT-LBS
BOTH RUDDERS                           -104.45749    +819.62649                                            -1564420.0
BOTH PROPELLER DUCTS ................                +1072.2153                                            -071945.63
BOTH THRUST NOZZLES                    +6135.4381    -5952.1256                  +17946.375    -18406.312    +249814.87
AERODYNAMICS-WINDAGE ................  -2853.2684    +2536.9794                  -20295.835    +51357.748    -152266.44
AERODYNAMIC DAMPING                                                                                         -099438.00
AIR MOMENTUM ........................  -7878.6060    +1205.7851                  -9646.2416    -60628.841    -154119.00
CUSHION PRESSURE                                                   -370210.93    +6724170.8    -10840774.
GRAVITY .............................  +6157.1568    -43313.436    +349993.49    -5953374.8    +10549061.    +1410231.1
SKIRT & SPRAY DRAG                     -1371.6875    +34.725252                  -579.82083    -22903.597    -25737.128
SEAWAY & VEHICLE GENERATED WAVES ...   -7365.5927    -9557.1957                                              +262201.25
VEHICLE GENERATED WAVES
TOTAL--SUMMATION OF ABOVE ..........   +10571.914    -54822.669    -20217.437    +722378.10    -342294.99    +1748072.1  LBS/FT-LBS
RESIDUE & CROSS-COUPLING               -5895.9225    -37971.764    -25882.120    -1033924.7    +856966.90    -1992503.1
TOTAL ..............................   +4675.9929    -092794.46    -46099.557    -311598.68    +514671.92    -244431.24  LBS/FT-LBS

ACCELERATIONS:                         ****************************** ACCELERATIONS ****************************
UDOT,VDOT,WDOT,PDOT,QDOT,RDOT ......   +.2979879     -8.5292558    -4.2372866    +.05493630    +.03159435    -.01188290  FT,RAD/S/S
VELOCITIES:                            ******************************* VELOCITIES *****************************
INITIAL.....UC,VC,WC,        RC ......  +.00000000    +.00000000    +.00000000                                +.00000000  FT/S,RAD/S
INTEGRATED..U',V',W',P',Q',R'          +74.094635    -12.213622    +2.5978461    +.01786344    -.00912950    +.03531140
TOTAL.......U ,V ,W ,P ,Q ,R .......   +74.094635    -12.213622    +2.5978461    +.01786344    -.00912950    +.03531140  FT/S,RAD/S
                                       ****************************************************************************
                                         NORTH         EAST         VERTICAL       ROLL          PITCH         HEADING
EARTH-FIXED COORDINATES                ****************************************************************************
VELOCITIES:                            ******************************* VELOCITIES *****************************
XI,YI,ZIDOT,PHI,THETA,PSIDOT .......   -72.474365    -19.477617    +.21713830    +.91790813    -.27270203    +2.0879249  FT/S,DEG/S
POSITIONS:                             ******************************* POSITIONS ******************************
INITIAL.....XI,YI,ZIC,          PSIO   +29999.999    +.00000000    -16.679995                                +.00000000  FT,DEG
INTEGRATED..XI,YI,ZI',PHI,THETA,PSI'   -15643.582    -2006.2349    +.14750150    +7.0906390    +1.0079595    +204.64312
TOTAL.......XI,YI,ZI ,PHI,THETA,PSI    +45643.580    +2006.2349    -16.532381    +7.0906390    +1.0074455    -155.35586  FT,DEG
PHIRAD,THETARAD,PSIRAD                                                           +.12375499    +.01759221    +2.7114888  RAD

EFFECTORS:                             ******************************* EFFECTORS ******************************
NPS,NPP,DELTAS ,DELTAP ,PSIS ,PSIP     +1344.1350    +1341.0905    +23.665957    +26.400131    -3.9189703    -44.277024  RPM,DEG
THRUST(S,P),DELTARC(S,P),PSINC,PSINCA  +8086.2075    +9467.1098    +23.665957    +26.400131    -3.9187749    -71.787964  LBS,DEG
CUSHION & MANIFOLD PRESSURES 1-4,S-P   +119.25282    +108.35874    +114.72758    +120.42785    +146.04391    +148.23037  LBS/FT**2
AIR FLOW: CFANS,CFANP,CNOZS,CNOZP      +21699.335    +21454.562    +4178.3952    +4210.4030                              FT**3/SEC
FAN SPEEDS: STBD,PORT                  +2091.3881    +2036.6511                                                            RPM
CUSHION VOLUME .....................   +2910.7433    +3475.7755    +2552.3615    +934.42407                              FT**3
WAVE PUMPING AIR FLOW                  +113.26658    -309.15507    -390.69804    -153.58396                              FT**3/SEC
SKIRT GAP AREA .....................   +138.75469    +527.01224    +133.26873    +.00000000                              FT**2
GAS TURBINE SPEED                      +18006.019    +18006.019    +18006.019    +18188.574    +18188.574    +18188.574  RPM
POWER TURBINE SPEED ................   +16124.812    +16086.289                                                            RPM
GAS TURBINE COMMAND                    +18512.976    +18512.976    +18512.976    +18512.976    +18512.976    +18512.976  RPM
POWER TURBINE COMMAND ..............   +16000.000    +16000.000                                                            RPM

ENVIRONMENT:                           ****************************** ENVIRONMENT *****************************
WIND AND SET .......................   +.00000000    +.00000000                                                            FT/SEC,DEG
SEAWAY HEIGHT,PERIOD,SLOPE             +.00000000    +9.9997999    +.02000000                                              FT,SEC,---
NORMAL, OFFSHORE LAMBDA, OMEGAE ....   -12802.621    +512.00126    +.17851902                                              FT,RAD/S
SYSTEM PARAMETERS:                     *************************** SYSTEM PARAMETERS **************************
TIME,CYCLE TIME,TIME OVERFLOW ......   +532.14999    +20484        +0                                                      SEC,USEC,-
```

Figure 53. ACV System Printout - Sample

PARAMETER VALUES FOR INITIALIZATION

U0	Velocity (linear) in direction X_b (Ft/sec)
V0	Velocity (linear) in direction Y_b (Ft/sec)
W0	Velocity (linear) in direction Z_b (Ft/sec)
P0	Velocity (angular) about axis X_b (rad/sec)
Q0	Velocity (angular) about axis Y_b (rad/sec)
R0	Velocity (angular) about axis Z_b (rad/sec)
XI 0	Inertial position (North) of pilot cabin (Ft)
YI 0	Inertial position (East) of pilot cabin (Ft)
ZI0	Inertial position (Vertical) of pilot cabin (Ft)
PHI 0	Euler Angle Roll (Deg)
THETA 0	Euler Angle Pitch (Deg)
PSI 0	Euler Angle Heading (Deg)
WIND	Wind Velocity (Ft/sec)
PSIWIND	Set of Wind (Deg)
DELTAT	Sampling Period (Sec)
PCUSH10	Pressure within cushion compartment 1 (Lbs/Ft^2)

PCUSH20	Pressure within cushion compartment 2 (Lbs/Ft^2)
PCUSH30	Pressure within cushion compartment 3 (Lbs/Ft^2)
PCUSH40	Pressure within cushion compartment 4 (Lbs/Ft^2)
PMANS0	Pressure within STBD manifold (Lbs/Ft^2)
PMANP0	Pressure within PORT manifold (Lbs/Ft^2)
NTS10	Gas Turbine Speed - STBD 1 (RPM)
NSSØ	Power Shaft Speed - STBD 1 (RPM)
VGWFLAG	Set = 1 to run VGW
BETAPSØ	Propeller Pitch - STBD (Deg)
BETAPPØ	Propeller Pitch - PORT (Deg)
PSINØ	Thrust Nozzle Angle (Deg)
PERIOD	Offshore Wave Period (Sec)
SLOPE	Ocean Bottom Slope
HEIGHT	Offshore Wave Height (Ft)
CSKIRT	Skirt Stiffness Coefficient

SUMMARY OF ACV OPERATING PROCEDURES

System Sense Switches (SSW)

1. RESET (Simulation mode)
2. FREEZE (Simulation mode)
3. OPERATE (Simulation mode)
8. Read Cards
9. Print State Parameters
11. Time Background (Units of 12 μ secs)
15. Print Datapool Contents (FS)

Console Sense Switches (CSW)

1. Inhibit ACV Simulation (Do not run program ACVSIM)
2. Inhibit ACV Cockpit Command Inputs
3. Inhibit CRT Driver (DLOOP)

Console Interrupt

Alternately allow or disallow type-ins

SUMMARY OF ACV OPERATING PROCEDURES (CONTINUED)

Traps

Error Locations

C4ERR#	- right-most hex character equal to trap error number
C4ERRLOC	- PSW1 of trap PSD
C4ERRMEM	- memory module fault bits (24-31)

Trap Error Number

0	Memory Parity
1	Non-allowed Operation
2	Unimplemented Instruction
3	Push Down Stack Fault
4	Fixed Point Arithmetic Overflow
5	Floating Point Fault
6	Decimal Arithmetic Fault
7	Watch Dog Timer

Trap Restarts

1.	CSW1 Off-IDLE-RUN	(pick up from error)
2.	CSW1 On-IDLE-RUN	(reinitialize - only after trap)
3.	IDLE-CPU RESET-RUN	(reinitialize - anytime)

SUMMARY OF ACV OPERATING PROCEDURES (CONTINUED)

Card Inputs

ACVINIT		Start card processing
MEMORY	ADDRESS, DATA, DATA, . . .	Absolute memory alteration
PARAM	PARAMETER, DATA, DATA, . . .	Parametric memory alteration
ENDACVIN		End card processing

(SSW 8 can be put off after ACVINIT is read. Card reading will continue until ENDACVIN is read)

Type Inputs

TIME	HH:MM	Set time (HH = 0-23, MM = 0-59)
DATE	MON DD, 'YY	Set date (MON = JAN, FEB, etc. . . , one space, DD = 01-31, 'YY = '74, '75, '76)
MEMORY	ADDRESS, DATA, DATA, . . .	Absolute memory alteration
PARAM	PARAMETER, DATA, DATA, . . .	Parametric memory alteration
DUMP D	DATA	Dump integer in decimal
DUMP X	DATA	Dump integer in hex
DUMP F	DATA	Dump floating point word as integer-fraction
DUMP E	DATA	Dump floating point word as mantissa-exponent

Type Control Keys

NL	End and process message
HT	End and process message
EOM	End and ignore message
85 characters	End and ignore message
n¢	Ignore last n characters

SUMMARY OF ACV OPERATING PROCEDURES (CONTINUED)

Input DATA Format

MMM D'±DDDDDD'	Decimal integer B31
MMM D'±.DDDDD'	Decimal fraction B0
MMM X'HHHHHH'	Hexadecimal integer
MMM F'±D.DDDDDE±DD'	Floating point single value

MMM - represents a multiple for consecutive locations stored or dumped. It is assumed to be 1 if not given.

ADDRESS Format

The same as DATA but the value is considered a memory address and the multiple (MMM) is ignored, if given.

SECTION VI

DERIVATIONS

Those equations (sections of the model) that changed to a degree that might not be recognizable from those shown in the report of the math model, the derivations are shown in this section. The only model that does not either coincide with the math model report or is not shown here is the compartment pressure model. The air pressures are solved in a similar manner (that is shown), however, the derivation was performed by Damon Cummings.

CHANGE IN INERTIAL POSITION

This derivation was performed on the second order predictor-corrector integration scheme to integrate velocity to a change in inertial X and Y position for every sampling period. This was done to improve accuracy on a delta X calculation, subtracting one large value from another large value.

$$P_t = X_{t-2\Delta t} + 2\Delta t\, \dot{X}_{t-\Delta t}$$

$$C_t = X_{t-\Delta t} + \frac{\Delta t}{2} (\dot{X}_t + \dot{X}_{t-\Delta t})$$

$$\Delta PC = 4/5 \ (C_t - P_t)$$

$$\begin{aligned} X_t &= P_t + \Delta PC \\ &= P_t + 4/5\ (C_t - P_t) \\ &= C_t + 1/5\ (P_t - C_t) \\ &= X_{t-\Delta t} + \frac{\Delta t}{2} (\dot{X}_t + \dot{X}_{t-\Delta t}) + 1/5 \Big[X_{t-2\Delta t} + 2\Delta t\, \dot{X}_{t-\Delta t} \\ &\quad - X_{t-\Delta t} - \frac{\Delta t}{2} (\dot{X}_i + \dot{X}_{t-\Delta t}) \Big] \end{aligned}$$

$$\begin{aligned} X_t - X_{t-\Delta t} &= 1/2\, \Delta t\, \dot{X}_t + 1/2\, \Delta t\, \dot{X}_{t-\Delta t} - 1/5\ (X_{t-\Delta t} - X_{t-2\Delta t}) \\ &\quad + 2/5\, \Delta t\ \dot{X}_{t-\Delta t} - 1/10\, \Delta t\, \dot{X}_i - 1/10 \Delta t\, \dot{X}_{t-\Delta t} \end{aligned}$$

$$\begin{aligned} \Delta X_i &= 4/5\, \Delta t\, \dot{X}_{t-\Delta t} + 2/5\, \Delta t\, \dot{X}_t - 1/5\, \Delta X_{t-\Delta t} \\ &= 1/5 \Big[2\, \Delta t\ (2\, \dot{X}_{t-\Delta t} + \dot{X}_t) - \Delta\, X_{t-\Delta t} \Big] \end{aligned}$$

SEAWAY

The wind generated waves have complex equations in the shoaling region for the first order amplitude and the second order Stokes' amplitude. Both equations taken from the math model contained hyperbolic functions. The hyperbolic identifies and the wave number equation were substituted appropriately to obtain simpler equations that would decrease computation time.

Reduction of the complex shoaling equation at the deep ocean boundary is shown to vastly simplify the offshore wave equation.

$$H_x = H \left[\frac{2 \cosh^2 K_x D_x}{2 K_x D_x + \sinh 2 K_x D_x} \right]^{1/2}$$

$$= H \left[\frac{\dfrac{2}{1-\tanh^2 K_x D_x}}{2K_x D_x + 2 \dfrac{\tanh K_x D_x}{(1-\tanh^2 K_x D_x)^{1/2}} \quad \dfrac{1}{(1-\tanh^2 K_x D_x)^{1/2}}} \right]^{1/2}$$

$$= H \left[\frac{1}{K_x D_x \; (1-\tanh^2 K_x D_x) + \tanh K_x D_x} \right]^{1/2}$$

$$= H \left[\frac{1}{K_x D_x - \dfrac{K_o^2 D_x}{K_x} \quad + \dfrac{K_o}{K_x}} \right]^{1/2}$$

$$= H \left[\frac{K_x}{K_x^2 D_x - K_o^2 D_x + K_o} \right]^{1/2}$$

$$\eta_2 = \frac{\pi}{8} \frac{H_x^{\;2}}{\frac{2\pi}{K_x}} \frac{\left[\cosh K_x D_x\right]\left[2 + \cosh 2 K_x D_x\right]}{\left[\sinh^3 K_x D_x\right]} \leq \frac{H_x}{8}$$

$$= \frac{1}{16} K_x H_x^2 \frac{\left[\frac{1}{(1-\tanh^2 K_x D_x)^{1/2}}\right]\left[2 + \frac{1 + \tanh^2 K_x D_x}{1 - \tanh^2 K_x D_x}\right]}{\left[\frac{\tanh^3 K_x D_x}{(1-\tanh^2 K_x D_x)^{3/2}}\right]} \leq \frac{H_x}{8}$$

$$= \frac{1}{16} K_x H_x^2 \frac{\left[1-\tanh^2 K_x D_x\right]^2\left[3-\tanh^2 K_x D_x\right]}{\left[\tanh^3 K_x D_x\right]} \leq \frac{H_x}{8}$$

$$2\eta_2 = \frac{1}{8} K_x H_x^2 \frac{\left[1-\left(\frac{K_o}{K_x}\right)^2\right]\left[3-\left(\frac{K_o}{K_x}\right)^2\right]}{\left(\frac{K_o}{K_x}\right)^3} \leq \frac{H_x}{4}$$

$$= H_x^{\;2}\left[\frac{3}{8}\frac{K_x^{\;4}}{K_o^3} - \frac{7}{8}\frac{K_x^{\;2}}{K_o} - \frac{1}{8}\frac{K_o^{\;3}}{K_x^{\;2}} + \frac{5}{8} K_o\right] \leq \frac{H_x}{4}$$

$$\eta_{seak} = \eta_1 + \eta_2 \text{ Cos } 2 \, (\upsilon_x - \omega_o t)$$

$$= \frac{H_x}{2} \text{ Cos } (\upsilon_x - \omega_o t) + \frac{H_x^{\,2}}{2} \left[\frac{3}{8} \frac{K_x^{\,4}}{K_o^{\,3}} - \frac{7}{8} \frac{K_x^{\,2}}{K_o} - \frac{1}{8} \frac{K_o^{\,3}}{K_x^{\,2}} + \frac{5}{8} K_o \right] \text{ Cos } 2 \, (\upsilon_x - \omega_o t)$$

OFFSHORE

$$\eta_{sea\ k} = \frac{H}{2} \text{ Cos } (\upsilon_o - \omega_o t) + \frac{H^2}{2} \left[\frac{3}{8} \frac{K_o^{\,4}}{K_o^{\,3}} - \frac{7}{8} \frac{K_o^{\,2}}{K_o} - \frac{1}{8} \frac{K_o^{\,3}}{K_o^{\,2}} + \frac{5}{8} K_o \right] \text{ Cos } 2 \, (\upsilon_o - \omega_o t)$$

$$= \frac{H}{2} \text{ Cos } (\upsilon_o - \omega_o t) + \frac{H^2}{2} \left[\frac{3}{8} K_o - \frac{7}{8} K_o - \frac{1}{8} K_o + \frac{5}{8} K_o \right] \text{ Cos } 2 \, (\upsilon_o - \omega_o t)$$

$$= \frac{H}{2} \text{ Cos } (\upsilon_o - \omega_o t)$$

$$= \frac{H}{2} \text{ Cos } \left[K_o \, (X_{iHULLk} - X_S) - \omega_o t \right]$$

SECTION VII

HULL BOTTOM PLATING - 35 POINTS

This section shows the equations in present form of the original set from the "Mathematical Model of an Air Cushioned Vehicle". These equations show the mathematics of the forces and moments due to the wave heights, cushion compartment volumes, and skirt escape areas for the original 35 points under the hull planform. The number of points was decreased in an attempt to get the VGW to run real-time. These equations are presented here for completeness only.

Three subroutines and the DATAPOOL changed as a result of changing from 35 points to 25 points. Other equations and subroutines depend upon these hull point positions and the numbering scheme, however, that is simply handled by changing the number of points and their locations within the DATAPOOL. See Figure 56 and Table 12 for hull point definitions.

These equations necessitated a change as special numerical techniques were developed for solution purposes.

Subroutine: SEA

SEAWAY AND VEHICLE GENERATED WAVES (Forces & Moments)

$$
\begin{aligned}
X_{SEA} = -\frac{20}{3}\Big[& P_{CUSH\,1}\,(\eta_1+\eta_2+\eta_3-\eta_6-\eta_7-\eta_8) \\
& +P_{CUSH2}\,(\eta_6+\eta_7+\eta_8-\eta_{21}-\eta_{27}-\eta_{26}) \\
& +P_{CUSH3}\,(\eta_8+\eta_{16}+\eta_{15}-\eta_{19}-\eta_{20}-\eta_{21}) \\
& +P_{CUSH4}\,(\eta_1+\eta_{11}+\eta_{12}-\eta_8-\eta_{16}-\eta_{15})\Big]
\end{aligned}
$$

$$
\begin{aligned}
Y_{SEA} = -\frac{40}{4}\Big[& P_{CUSH\,1}\,(\eta_3+\eta_4+\eta_5+\eta_6-\eta_1-\eta_{10}-\eta_9-\eta_8) \\
& +P_{CUSH2}\,(\eta_6+\eta_{24}+\eta_{25}+\eta_{26}-\eta_8-\eta_{23}-\eta_{22}-\eta_{21}) \\
& +P_{CUSH3}\,(\eta_{21}+\eta_{22}+\eta_{23}+\eta_8-\eta_{15}-\eta_{17}-\eta_{18}-\eta_{19}) \\
& +P_{CUSH\,4}\,(\eta_8+\eta_9+\eta_{10}+\eta_1-\eta_{12}-\eta_{13}-\eta_{14}-\eta_{15})\Big]
\end{aligned}
$$

$$
\begin{aligned}
N_{SEA} = P_{CUSH1} & \Big[\frac{(20)(8)}{3}\,(\eta_6+\eta_7+\eta_8-\eta_1-\eta_2-\eta_3) \\
& +\frac{(40)(10)}{4}\,(\eta_6+\eta_5+\eta_4+\eta_3-\eta_8-\eta_9-\eta_{10}-\eta_1)\Big] \\
+P_{CUSH2} & \Big[\frac{(40)(50)}{4}\,(\eta_{26}+\eta_{25}+\eta_{24}+\eta_6-\eta_{21}-\eta_{22}-\eta_{23}-\eta_8) \\
& +\frac{(20(8)}{4}\,(\eta_{21}+\eta_{27}+\eta_{26}-\eta_8-\eta_7-\eta_6)\Big] \\
+P_{CUSH3} & \Big[\frac{(20)(28)}{3}\,(\eta_{19}+\eta_{20}+\eta_{21}-\eta_{15}-\eta_{16}-\eta_8) \\
& +\frac{(40)(50)}{4}\,(\eta_{21}+\eta_{22}+\eta_{23}+\eta_8-\eta_{19}-\eta_{18}-\eta_{17}-\eta_{15})\Big] \\
+P_{CUSH4} & \Big[\frac{(20(28)}{3}\,(\eta_8+\eta_{15}+\eta_{16}-\eta_1-\eta_{11}-\eta_{12}) \\
& +\frac{(40)(10)}{4}\,(\eta_8+\eta_9+\eta_{10}+\eta_1-\eta_{12}-\eta_{13}-\eta_{14}-\eta_{15})\Big]
\end{aligned}
$$

i	1-27, Cushion periphery points
η_i	Total height of waves due to Seaway & VGW(Ft)
$P_{CUSH\ 1}$	Pressure in cushion compartment 1 (Lbs/Ft^2)
$P_{CUSH\ 2}$	Pressure in cushion compartment 2 (Lbs/Ft^2)
$P_{CUSH\ 3}$	Pressure in cushion compartment 3 (Lbs/Ft^2)
$P_{CUSH\ 4}$	Pressure in cushion compartment 4 (Lbs/Ft^2)

Subroutine: VOLCUSH

$$H_{CUSH\ 1} = .020833\ (h_{w1} + h_{w3} + h_{w6} + h_{w8}) + .0625\ (h_{w4} + h_{w5} + h_{w9} + h_{w10}) + .08333\ (h_{w2} + h_{w7}) + .25\ (h_{w30} + h_{w31})$$

$$H_{CUSH\ 2} = .020833\ (h_{w6} + h_{w26} + h_{w21} + h_{w8}) + .0625\ (h_{w22} + h_{w23} + h_{w24} + h_{w25}) + .08333\ (h_{w7} + h_{w27}) + .25\ (h_{w28} + h_{w29})$$

$$H_{CUSH\ 3} = .020833\ (h_{w8} + h_{w21} + h_{w19} + h_{w15}) + .0625\ (h_{w17} + h_{w18} + h_{w22} + h_{w23}) + .08333\ (h_{w16} + h_{w20}) + .25\ (h_{w34} + h_{w35})$$

$$H_{CUSH\ 4} = .020833\ (h_{w1} + h_{w8} + h_{w12} + h_{w15}) + .0625\ (h_{w9} + h_{w10} + h_{w13} + h_{w14}) + .08333\ (h_{w11} + h_{w10}) + .25\ (h_{w32} + h_{w33})$$

$$V_{CUSH\ n} = A_{CUSH} H_{CUSH\ n}$$

n = 1---4	Number of Cushion Compartments
h_{wk}	Height of hull points over water (Ft)
A_{CUSH}	Cushion compartment area = 800 Ft^2
$H_{CUSH\ n}$	Average Height above Water of Cushion Compartment (Ft)
$V_{CUSH\ n}$	Air Volume of Cushion Compartment (Ft^3)
k = 1---35	Number of Hull Bottom Points

SUBROUTINE: SKAREA

CUSHION COMPARTMENT ESCAPE AREA

$$CLR_k = h_{wk} - h_s \geq 0$$

$$AREA_1 = 62.76\,[CLR_1 + 4\,CLR_2 + CLR_3]$$
$$+\,94.16\,[CLR_3 + 3\,CLR_4 + 3\,CLR_5 + CLR_6]$$

$$AREA_2 = 62.76\,[CLR_{26} + 4\,CLR_{27} + CLR_{21}]$$
$$+\,94.16\,[CLR_6 + 3\,CLR_{24} + 3\,CLR_{25} + CLR_{26}]$$

$$AREA_3 = 62.76\,[CLR_{21} + 4\,CLR_{20} + CLR_{19}]$$
$$+\,94.16\,[CLR_{19} + 3\,CLR_{18} + 3\,CLR_{17} + CLR_{15}]$$

$$AREA_4 = 62.76\,[CLR_{12} + 4\,CLR_{11} + CLR_1]$$
$$+\,94.16\,[CLR_{12} + CLR_{13} + 3\,CLR_{14} + CLR_{15}]$$

$k = 1 - - - 27$	Number of Cushion Periphery Points
h_{wk}	HEIGHT OF HULL POINTS OVER WATER (FT)
h_s	HEIGHT OF SKIRT = 4.5 FT
CLR_k	AIR GAP BETWEEN SKIRT AND WATER OF HULL POINTS (FT)
$AREA_1$	CUSHION COMPARTMENT 1 ESCAPE AREA (FT^2) • Constant
$AREA_2$	CUSHION COMPARTMENT 2 ESCAPE AREA (FT^2) • Constant
$AREA_3$	CUSHION COMPARTMENT 3 ESCAPE AREA (FT^2) • Constant
$AREA_4$	CUSHION COMPARTMENT 4 ESCAPE AREA (FT^2) • Constant

Cushion Compartments & Hull Bottom Points

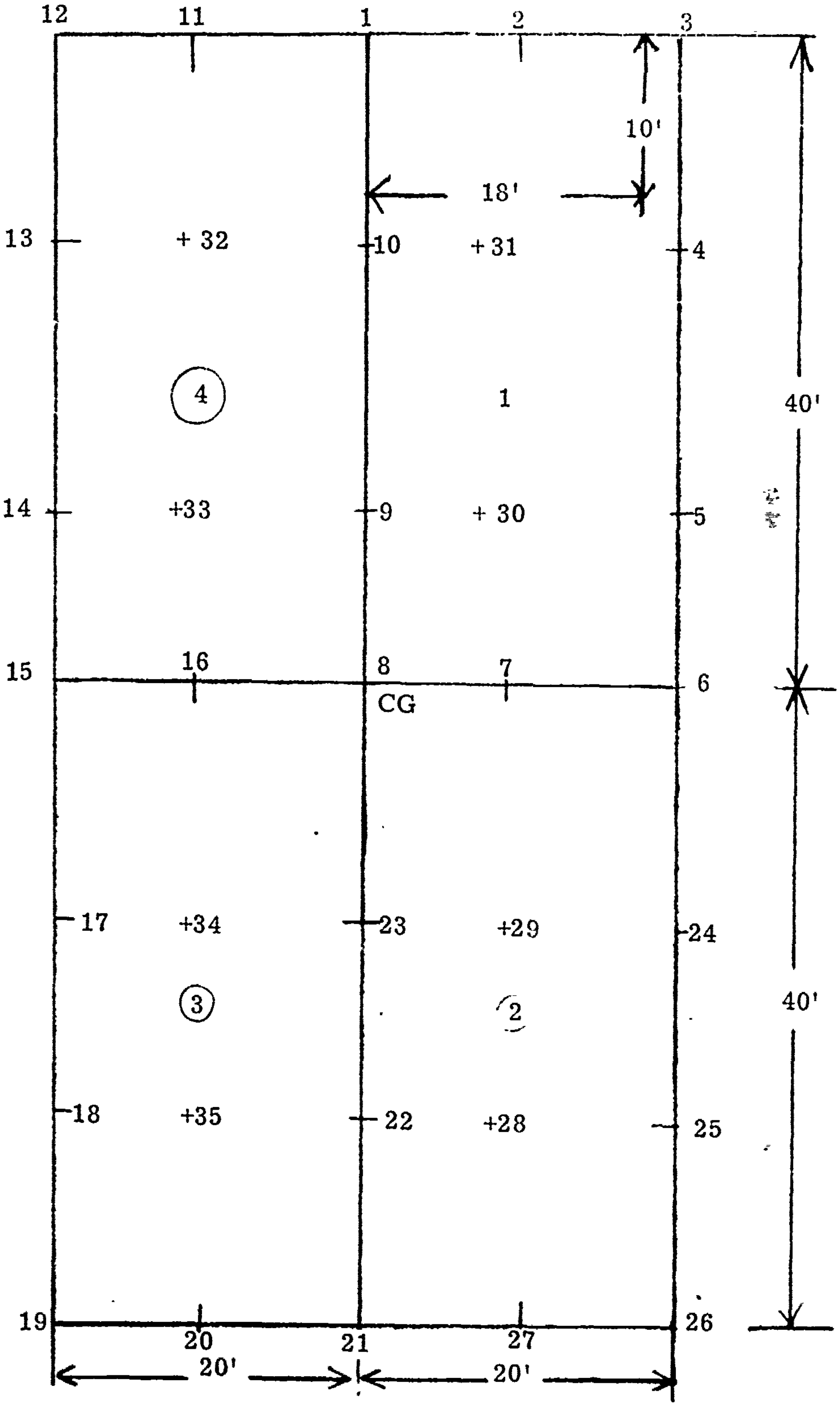


Figure 56. Hull Planform Profile (35 Pts)

TABLE 12. PLANFORM POINT LOCATIONS (32)

k	X_{HULL}	Y_{HULL}	Z_{HULL} (FT)
1	10	-18	12
2	10	-8	12
3	10	2	12
4	-3 1/3	2	12
5	-16 2/3	2	12
6	-30	2	12
7	-30	-8	12
8	-30	-18	12
9	-16 2/3	-18	12
10	-3 1/3	-18	12
11	10	-28	12
12	10	-38	12
13	-3 1/3	-38	12
14	-16 2/3	-38	12
15	-30	-38	12
16	-30	-28	12
17	-43 1/3	-38	12
18	-56 2/3	-38	12
19	-70	-38	12
20	-70	-28	12
21	-70	-18	12
22	-56 2/3	-18	12
23	-43 1/3	-18	12

TABLE 12. PLANFORM POINT LOCATIONS (35)(CONT'D)

k	X_{HULL}	Y_{HULL}	Z_{HULL}	(FT)
24	-43 1/3	2	12	
25	-56 2/3	2	12	
26	-70	2	12	
27	-70	-8	12	
28	-56 2/3	-8	12	
29	-43 1/3	-8	12	
30	-16 2/3	-8	12	
31	-3 1/3	-8	12	
32	-3 1/3	-28	12	
33	-16 2/3	-28	12	
34	-43 1/3	-28	12	
35	-56 2/3	-28	12	

SECTION VIII

I/O PROGRAM WRITEUPS

The basic Background Task Input/Output programs are individually described. Also, the trap handling routines are individually described. The program name, type, and purpose are shown. The calling program is named and the proper calling sequence of the I/O program is shown. The program Description includes the definition of calling sequence and the functional ability of each individual program.

The list of program writeups is as follows:

ATD
BACKGRND
BCDATA
BCDSUBS - BCDCOMMA, BCDPARAM, BCDSTORE
CARDS, CARDIO
DISCRETE
DLOOPER
DTA
EXECACV
FIXPRNT
FLTDEC
FLTPRNT
INTEGER - DECIT, HEXIT
IOPERIF - CARDREAD, PRNTLINE, PRNTPAGE, TYPEOUT, TYPEIN
PRINT
PRNTTIME
TYPE
TYPEMSG

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	ATD
PROGRAM TYPE:	Analog Input Driver
INITIATED BY:	Program ACVSIM
PURPOSE:	To read and format analog input data
CALLING SEQUENCE:	BAL, 15 ATD

DESCRIPTION:

The ATD program reads all the analog input channels used by the system. It scales the values to engineering units and adds a bias, if one applies. The input value is finally limited to predefined maximums and minimums and stored in the simulation data values.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	BACKGRND
PROGRAM TYPE:	Background Timer
INITIATED BY:	Program PRINT
PURPOSE:	To provide timing for the background and indirectly the real time simulation.
CALLING SEQUENCE:	B BACKGRND

DESCRIPTION:

The program BACKGRND will determine how long the background idle loop is in operation. This value is used to determine the execution time of the real time programs. BACKGRND increments a counter which is scaled at 12 μsec per unit. The timing is dependent on the instructions used in the increment loop. The timing is done only at the option of the operator. The time value is stored in location BGTIMSAV by the real time interrupt, EXECACV. If the timing is stopped by the operator BACKGRND is terminated by branching to the program CRT.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: BCDATA

PROGRAM TYPE: I/O Subroutine

INITIATED BY: EBCDIC Input Routines

PURPOSE: Translate EBCDIC character strings into one word, numeric values.

CALLING SEQUENCE:

```
LW,6  BA(STRTADDR)
LW,7  BA(ENDADDR)
BAL,15 BCDATA
STW,8 WORD
STW,5 MULT
```

DESCRIPTION:

The subroutine BCDATA will translate a specified byte field from EBCDIC character to a numeric word value. The values STRTADDR and ENDADDR are the starting and ending byte addresses of the field to be scanned. The output word value (WORD) will be in register 8 when the caller resumes. A value called the "multiple" (MULT) will be in register 5. This word will be explained below. The contents of register 9 will be destroyed.

The input format can be one of the following:

MMM X'HHHHHH'	Hexadecimal integer
MMM D'±DDDDDD'	Decimal integer
MMM D'± .DDDDDD'	Decimal fraction
MMM D'±D. DDDDDE±DD'	Floating point, single value

The value MMM is optional. If included, it is translated as a decimal integer and is returned in register 5. This value is used to specify how many times the value that follows it will be used by the calling program. It is, therefore called a "multiple". If the multiple is not present, the value 1 will be placed in register 5.

The format of the remainder of the field is described in the ACV Console Operating Procedures and in the program listing for BCDATA. They will not be repeated in detail here. Suffice it to say that the hexidecimal format is translated to a hexidecimal integer value right adjusted. The decimal integer format is translated to a decimal integer value scaled B31. The decimal fraction format is translated to a decimal fraction scaled B0. The floating point format is represented as a tens power, mantissa-exponent value. It is translated to a standard hex power, single word, floating point value.

If any irregularity in the format is found that cannot be translated, a -1 is placed in the return value of register 5. No value is resolved.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	BCDSUBS
PROGRAM TYPE:	Input Utility Subroutines
INITIATED BY:	The Programs TYPE and CARDS
PURPOSE:	To Decode input EBCDIC Data

CALLING SEQUENCE:

```
LI,6  BA (STRTADDR)
LI,8  BA(ENDADDR)
BAL,15  BCDCOMMA
STW,7  COMMADDR


LI,6  BA(STRTADDR)
LI,7  BA(ENDADDR)
BAL,15 BCDPARAM
STW,5  ADDRESS


LI,6  BA(STRTADDR)
LI,8  BA(ENDADDR)
LI,5  CODE
BAL,15  BCDSTORE
STW,5  ERROR
```

DESCRIPTION:

The program BCDSUBS is made up of three subroutines. These subroutines are used to decode EBCDIC character strings from the typewriter and the card reader.

The subroutine BCDCOMMA has the sole function of locating the character "," in a character buffer. The caller must specify the starting and ending byte location of the buffer (STRTADDR and ENDADDR, respectively). The address of the byte previous to the first comma encountered, when scanning from left to right, is stored in register 7 (COMMADDR). This specifies a field that contains valid input data.

There are two alternate conditions specified by the subroutine. If no comma is found but non-blank characters are found, a-2 is placed in register 7. This indicates the final field in the buffer. If no comma and no non-blank characters are found, a-1 is placed in register 7. This indicates either that a continuation line follows or that an error has occurred.

The subroutine BCDPARAM will search a specified byte field for a valid parameter that can be used on a PARAM input line. The byte field is specified by the byte starting address (STRTADDR) and the byte ending address (ENDADDR) in register 6 and 7, respectively. If a valid parameter is found, its associated memory word address is placed in register 5 (ADDRESS). If no valid parameter is found, register 5 will contain a-1.

The subroutine BCDSTORE translates the address and data fields from PARAM and MEMORY input byte strings and stores the data into the given address. The caller can specify one of three types of lines to translate. This value (CODE) is placed in register 5 by the caller. A 1 specifies a PARAM card. A 2 specified a MEMORY card. A 3 specifies a continuation card from a previous PARAM or MEMORY card. The byte starting address (STRTADDR) and the byte ending address (ENDADDR) of the input buffer are stored in register 6 and 8, respectively. When the caller of BCDSTORE is resumed, register 5 (ERROR) will be either 0 or -1. A 0 specifies a good input line. A -1 specifies that an error was found somewhere in the input line. The location of the error in the line is not specified.

The PARAM and MEMORY lines are processed by decoding the first field of the line. This should contain the starting address into which the proceeding data is to be stored. The address field must be on the same line that the keyword PARAM or MEMORY is on and must be followed by a comma. The data fields are then decoded and stored one after the other. The ending address +1 of the last data stored becomes the starting address of the next data. If a buffer is ended with a comma a-continuation line is assumed to follow. The flag (BCDSTWFL) is set to notify the caller of this fact. If the caller finds a continuation line (register 5=3), the final address of the last call will be used to store the new data.

If a data field cannot be decoded, the remainder of the input buffer is ignored and register 5 will be set to -1. Any stores that were made up to the point of the error will stay in effect. The continuation flag ,BCDSTWFL, will remain reset.

If the input line was intended to have a continuation line, that line will be considered an error, since it will not contain a valid keyword. Of course, if the address field contained an error, register 5 will be set to -1, but no memory stores will be performed.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: CARDS

PROGRAM TYPE: Background I/O

INITIATED BY: Program TYPEMSG or TYPE

PURPOSE: To read and process card reader input cards

CALLING SEQUENCE: B CARDS

DESCRIPTION:

The program CARDS reads cards on the request from the operator. If system sense switch 8 is set, cards will be read and printed directly onto the line printer. The program is called by either TYPE or TYPEMSG. The program will call the printer processor PRINT at the end of each card read.

The program will first look for the keyword, ACVINIT, starting on column one of each card read. Cards are bypassed until this card is read. When an ACVINIT card is read, only one of the following cards are valid: a PARAM card, a MEMORY card, or an ENDACVIN card. Also, at this point system sense switch 8 can be put off and cards will still be read until an ENDACVIN card is read. No cards will be processed after this card is read. If sense switch 8 is off when a valid ENDACVIN card is read, card reading will stop.

The PARAM and the MEMORY cards allow the computer memory to be altered. The format of these cards is discussed in the program listing and in the Console Operating Procedures. The details of their formats will not be repeated here. These cards can have continuation cards. Continuation is indicated by a comma as the last character on the card. Only if none of the three valid keywords are read and a continuation is indicated from the previous card, will the card be treated as a continuation card. Therefore, a card ending with a comma does not have to have a continuation card after it.

Any kind of format error will cause the error word and the remainder of the card to be bypassed and an error message to be printed on the line printer and the typewriter. Card processing will continue, however. That portion of the card before the error was found will be processed and any valid memory alteration that could be done will be done. Continuation cards, after an error card, will also be considered errors.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: DISCRETE

PROGRAM TYPE: Discrete Driver

INITIATED BY: Program ACVSIM

PURPOSE: To process the discrete I/O bits

CALLING SEQUENCE: BAL, 15 DISCRETE

DESCRIPTION:

The program DISCRETE reads the discrete bank used by the ACV system. The program will not return to the caller until all the discrete I/O is completed.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: DLOOPER

PROGRAM TYPE: CRT Driver

INITIATED BY: Prcgram CRT

PURPOSE: To set up ACV position variables for the CRT driver program DLOOP

CALLING SEQUENCE: BAL, 15 DLOOPER

DESCRIPTION:

The program DLOOPER converts the ACV simulation X, Y, Z position values and the sines and cosines of the pitch, roll, and heading from floating point values to integers and fractions. These conversions are made for a call to the CRT driver DLOOP. The X, Y, Z position values are converted into integer inches. The sines and cosines are converted into fractional values scaled B1. Then the program DLOOP is called to drive the CRT.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: DTA

PROGRAM TYPE: Analog Output Driver

INITIATED BY: Program ACVSIM

PURPOSE: To format and transmit variables to analog devices.

CALLING SEQUENCE: BAL, 15 DTA

DESCRIPTION:

The program DTA drives analog output devices such as the simulation cockpit display meters and the chart recorder. DTA takes the unformated floating point value directly from simulation parameters.

There are two types of analog output values. Those values that represent only a voltage and those that represent sines and cosines to a syncro meter. The syncro needs both the sine and cosine of its associated data value. Therefore, these values are processed twice.

The non syncro values are processed as follows. They are first adjusted to subtract any bias from them. Then their magnitude is limited so they don't go beyond the stops of their associated devices. Finally they are scaled to represent a fraction of 10 volts (10 volts is usually full scale on the output device). The formatted data value is stored in an analog output word for later transmittal.

Syncro values are first adjusted for any bias. They are then limited in magnitude to keep their value in meter limits. They are converted to degrees and compared to their last value. This comparison is to provide a meter change limit so that the meter is not changed to a new value too fast. Finally the sine or cosine of the syncro value is found and converted from floating point to fixed decimal B0. The value is stored in an analog output value for transmission.

When all the analog output values are converted, they are all sent out to the analog output converter.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	EXECACV
PROGRAM TYPE:	Interrupt Processor, Initialization
INITIATED BY:	Hardware Interrupts, Load Procedure
PURPOSE:	To process the interrupts and to initialize the ACV system
CALLING SEQUENCE:	(Interrupt Linkage)

DESCRIPTION:

The program EXECACV contains all the interrupt processors for the ACV system and also contains the initialization routine. The interrupts that are processed are the Counter 3 Zero Interrupt, the Control Panel Interrupts, the Memory Parity Interrupt, all the Error Traps, and the CAL4 Trap.

The Counter 3 Zero Interrupt provides the timing for the ACV simulation real time math model. It maintains the internal clock used to print out the time. It processes the background timer counter when the timing of the background is requested. The Counter 3 Zero can interrupt itself. If it does, a record of this happening is kept by incrementing a system flag. The real time programs are executed with Register Pointer set to 0, (Bits 55-59 of the PSD).

The ACV simulation is initialized at load time and any other time upon the request of the operator. During initialization the interrupt locations are reset, various system parameters are reset, and all I/O is reset. Location X' 26' is set to a branch to initialization routine. This allows the operator to restart the system by pressing IDLE-CPU RESET-RUN. The initialization routine is terminated by branching to background.

The Control Panel Interrupt does a logical exclusive or to a typewriter input flag. This will alternately set the flag to zero and non-zero. The flag is tested by the typewriter input I/O processor to allow and disallow the operator to perform input type commands.

The Power Up Interrupt will branch to ACV initialization. The Power Down Interrupt will execute an infinite loop.

The Memory Parity Interrupt and all the Error Traps are processed in the same way. They execute a CAL4 Trap. CAL4 will save the number of the error, the location and the status of the condition code at the time of the error, the status of the memory fault bits (for memory parity), and the 16 registers being used. The error can be recovered by either going back to the error location plus one or, if the console switch is on, going to the initialization routine.

EXECACV 2

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	FIXPRNT
PROGRAM TYPE:	Data Formatter
INITIATED BY:	Typewriter and line printer programs
PURPOSE:	To convert floating point values to EBCDIC fixed point values
CALLING SEQUENCE:	LI,2 BA(BUFFER) LW,6 DATA BAL,15 FIXPRNT

DESCRIPTION:

The program FIXPRNT converts a floating point, single value into EBCDIC print bytes. The format of the output is

SDDDDDDDDD

where S is the sign of the value, +or - . D is either a decimal character (0-9) or a decimal point. The position of the decimal point depends on the magnitude of the input value. If the value is smaller than ±10 ** -7 or larger than ± 10 **9 the data is converted to zero or ± 999999999, respectively. Leading zeros are suppressed. Ten characters are always stored in the output buffer (BUFFER).

The calling sequence expects register 2 to contain the byte address of the buffer in which to store the output bytes. Register 6 should contain the data value to be converted. When the calling routine is resumed, register 6 will be unchanged and register 2 will be incremented by 10.

The input data is converted from a single word, hexidecimal mantissa and exponent to a double word, decimal mantissa and exponent. The program FLTDEC is called to do this. The mantissa is then converted into EBCDIC characters using the exponent to position the decimal point.

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	FLTDEC
PROGRAM TYPE:	Data Conversion Subroutine
INITIATED BY:	Caller
PURPOSE:	To convert a single floating point value from a hexidecimal mantissa-exponent to a decimal mantissa-exponent
CALLING SEQUENCE:	LW,6 INDATA BAL,15 FLTDEC STW,6 OUTMANT STW,7 OUTEXP

DESCRIPTION:

The program FLTDEC converts standard single,floating point numbers from base 16 to base 10. The input value (INDATA) should be in register 6. The output mantissa (OUTMANT) and the output exponent (OUTEXP) are found in registers 6 and 7,respectively. The converted value is used to print floating point values onto the line printer and typewriter.

The method used is:

```
16 **E16=10**E10
Log (16**E16) = log (10**E10)
E16*log 16=E10
16**E16 = 10** (E16 * log 16)
```

E16** log 16 is broken up into its integer and fraction. We'll call them IE10 and FE10, respectively. Therefore:

```
MANT 16*(16**E16) = MANT16*(10**FE10)*(10**IE10)
```

Where MANT16 is the mantissa of the input floating point value. IE10 becomes the output exponent to the base 10.

MANT16*(10**FE10) is the new mantissa for the decimal floating point number. The decimal exponent and mantissa may get adjusted to insure that the mantissa is between +1 and -1.

The value of 10**FE10 is calculated with a Hastings polynomial. The coefficients are:

a_0 = 1.0

a_1 = 1.15129277

a_2 = .662739384

a_3 = .254393574

a_4 = .072951736

a_5 = .017421119

a_6 = .002554917

a_7 = .000932642

The polynomial is:

$$10**X = (\Sigma a_i X^i)^2$$

(See Approximations for Digital Computers by Cecil Hastings, Jr., Princeton University Press, 1955)

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	FLTPRNT
PROGRAM TYPE:	Data Formatter
INITIATED BY:	Typewriter and line printer programs
PURPOSE:	To convert floating point values to EBCDIC floating point values
CALLING SEQUENCE:	LI,2 BA(BUFFER) LW,6 DATA BAL,15 FLTPRNT

DESCRIPTION:

The program FLTPRNT converts floating point, single word data values into EBCDIC print bytes. The format of the output is:

SD.DDDDDDD ESDD

where S is the sign, + or -, of the mantissa and the exponent. The D's are decimal characters (0-9). The characters '.' and 'E' are constants. The exponent represents a power of ten exponent.

The calling sequence expects register 2 to contain the byte address of the buffer in which the output characters are to be stored. Register 6 contains the input floating point value. The calling program is resumed with register 6 unchanged and register 2 incremented by 16.

The input data is converted by the program FLTDEC from a hexidecimal mantissa-exponent to a double word, decimal mantissa - exponent. The double word is then converted into EBCDIC characters.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	INTEGER
PROGRAM TYPE:	Data Formatter
INITIATED BY:	Print and Typeout Programs
PURPOSE:	To convert computer words to decimal and hexi decimal integer EBCDIC formats.

CALLING SEQUENCE:

```
LI,2  BA(BUFFER)
LW,6  DATA
BAL,15  DECIT

LI,2  BA(BUFFER)
LW,6  DATA
BAL,15  HEXIT
```

DESCRIPTION:

The program INTEGER will format a data word into EBCDIC format. It will convert the value into EBCDIC decimal or hexi decimal characters depending on whether DECIT is called or HEXIT is called.

The word to be converted should be placed in register 6. The byte address of the buffer in which the formated bytes are to be stored, should be placed in register 2. In both cases the buffer address is updated by the number of bytes used in the formatting. HEXIT will always use eight bytes with leading zeros. DECIT suppresses leading zeros and left adjusts the characters. However, it will transfer trailing blanks and uses 11 bytes.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: IOPERIF

PROGRAM TYPE: I/O Subroutines Drivers

INITIATED BY: PRINT, TYPE, CARDS, TYPEMSG

PURPOSE: To provide the I/O interface between the hardware unit and its software processor.

CALLING SEQUENCE:

```
BAL,15  CARDREAD
DATA     BA(CARDBUF)

BAL,15  PRNTLINE
DATA  BA(PRNTBUF)

BAL,15  PRNTPAGE

BAL,15  TYPEOUT
DATA  BA(BUFFER)

BAL,15  TYPEIN
DATA   DA(BUFFER)
```

DESCRIPTION:

IOPERIF contains all the I/O drivers for the peripheral equipment. These subroutines are designed to be used in background only. The typewriter routines can be used by the typewriter processors only. Each subroutine is described below.

The routine CARDREAD will read one card and store the input characters in the buffer provided by the caller. The program will always store 80 characters. If a card reading error occurs, the program will wait for the operator to correct the error on the reader and reread a new or the corrected card.

The routine PRNTLINE will print one line of data onto the line printer. The input buffer should have as its first character the size of the remaining buffer. This first character is not printed. The input buffer can be any size but only 132 characters or less will be printed. When the calling program is reentered, it is free to change its buffer even though the last line is not completed printing.

The routine PRNTPAGE will skip the printer page to the head of page. No input buffers are needed. The next line printed will be put on the first line of the new page.

The routine TYPEOUT will type one line of data onto the typewriter. The first character of the input buffer should be equal to the size of the remaining buffer. The first character is not printed. The input buffer size should not be longer than 85 characters. The carriage return character is put at the end of the data line by the TYPEOUT subroutine. TYPEOUT returns to the calling program before the line is typed, but the calling buffer is immediately available for reuse. See the note below.

The subroutine TYPEIN reads one line of typed data. An input line on the typewriter is defined as one that ends with one of the three characters NL(new line), EOM (end of message), or HT (horizontal tab-TAB). A line that exceeds 85 characters will also be terminated as a complete typed input line. The line that was ended by an EOM character or because it was 85 characters long will be rejected by TYPEIN and TYPEIN will read a new line. Also if the line was terminated with an I/O error indication, it will be rejected and a new line will be read. The first character of the resultant input line will be equal to the size of the remaining buffer and should not be considered part of the input line.

The cent sign (¢) is used as a backspace by the operator. The input line is scanned for cent signs and TYPEIN will correct the line so that the caller will never see them. If the ¢ marks backspace the entire line, a new line will be read. See the note below.

Note - Since the typewriter is such a slow device especially on input, device busy tests use the routine TYPEWAIT. This routine will allow other devices besides the typewriter to be referenced even though the typewriter is in the middle of an I/O request. See the program TYPE.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: PRINT

PROGRAM TYPE: Background I/O

INITIATED BY: Program CARDS

PURPOSE: To print the state parameters

CALLING SEQUENCE: B PRINT

DESCRIPTION:

The program PRINT is a driver for the program SYSPRINT. SYSPRINT prints the simulation state parameters. If card reading is being done, printing will not be done. The program is terminated by branching to the program BACKGRND.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME:	PRNTTIME
PROGRAM TYPE:	I/O Subroutine
INITIATED BY:	SYSPRINT
PURPOSE:	To provide the time and date in EBCDIC format
CALLING SEQUENCE:	BAL, 15 PRNTTIME DATA BUFFER

DESCRIPTION:

The program PRNTTIME will store the EBCDIC time and date into the specified buffer (BUFFER in the calling sequence). The format of the time and date is:

HH:MM   MON DD, YY

HH is the military hours

MM is the minutes

MON is the 3 letter month

DD is the day of the month

YY is the last two numbers of the year

The data is stored on a word boundary and needs four words.

The time is calculated from the value RUNTIME which is updated by EXECACV. RUNTIME is divided by RUNFREQ which equals the driving frequency of the real time simulation. This calculates the elapsed time in seconds. The minutes and hours are easily derived from the elapsed seconds.

The hour is calculated to be between 0 and 23. If it gets to be 24 or larger, 24 hours is subtracted from the time and the day of the month is incremented. The month and the year are never changed. The day will be incremented past the proper number of days in the month up to a maximum of 99.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: TYPE

PROGRAM TYPE: Background I/O

INITIATED BY: Program CRT

PURPOSE: To read and process the typewriter input commands

CALLING SEQUENCE: B TYPE

DESCRIPTION:

The program TYPE will read and process typewriter input lines. The operator initiates an input line by putting on the typewriter input light. This is done with the Control Panel Interrupt. The interrupt alternately sets and resets the flag TYPEREQ. When the flag is non-zero (set), a read command is made to the typewriter control unit. It is this read command that causes the typewriter light to be put on. The operator then types a line onto the typewriter. The program will process it and perform a new read command. When the flag is zero (reset), the pending read command is cancelled and the light is turned off.

Because the typewriter is a slow input and output device, a routine has been implemented to allow other background tasks to be performed while the typewriter processors are waiting for the typewriter to become "not busy". This routine is TYPEWAIT. The routine is called during I/O operations by either of the routines TYPEOUT or TYPEIN. Both of these routines are in the program IOPERIF. TYPEWAIT saves all of the 16 registers and branches to the program CARDS. This allows other background I/O operations to be performed. Each time TYPE is entered a test is made to determine if TYPEWAIT was called. If it was, the routine that called TYPEWAIT is resumed with all of its registers intact.

The program TYPE is entered from the program CRT. It normally exits by branching to the program TYPEMSG. If TYPEWAIT is in process, TYPE will branch to CARDS.

If TYPEWAIT is not in effect, TYPE will follow its normal path as follows. If the ACV system was initialized, TYPE will first type a message to request the operator to enter the date and time. If a read request is indicated, a line is read and it is tested for a valid keyword. The keyword, which is the first word on a typed line, specifies the function to be performed. If a valid keyword is read, the line is processed by the keyword function processor. An invalid keyword which is not a continuation line (see MEMORY and PARAM) will cause an error line to be printed. If the function requested requires type-outs, the output operation is done in the program TYPE, not in TYPEMSG. When the line is completely processed or rejected as an error a new line is read until the operator resets the type request flag, TYPEREQ.

The TYPE program can recognize five keywords. They are TIME, DATE, MEMORY, PARAM, and DUMP. The format and function of each keyword request is in the program listing and the Console Operating Procedures. They will not be repeated here. MEMORY and PARAM are duplicates of the card reader inputs of the same names. They can have continuation lines. None of the other requests can. The time request will convert the input time into units of the real time update frequency. The time in seconds is multiplied by RUNFREQ and stored in RUNTIME. The date that is typed in with the DATE keyword is stored with no changes made to it. The DUMP request will space a line and type one location per line until all the requested locations are typed.

If any of the keyword processors cannot read the input line, an error line will be typed and the line will be ignored from the point of the error. Except for MEMORY and PARAM this means the whole line is ignored. The error is not localized by the software. The operator has to determine the location of the error in the typed line.

ACV

PROGRAM DESCRIPTION

(7/30/75)

PROGRAM NAME: TYPEMSG

PROGRAM TYPE: Background I/O

INITIATED BY: Program TYPE

PURPOSE: Typeout error messages onto the typewriter

CALLING SEQUENCE: B TYPEMSG

DESCRIPTION:

The program TYPEMSG tests positions on a message queue. If any message is flagged to be typed, it is typed. Because of the length of time it takes to type messages the queue is examined repeatedly until there are no messages left to type. The program ends by branching to the card reader I/O processor CARDS.

SECTION IX

SUGGESTIONS FOR CONTINUING DEVELOPMENT

As noted in the main body of this report, the Vehicle Generated Wave (VGW) model does not run in real-time. A considerable amount of work needs to go into simulation development to enhance real-time execution. As simple as the calculations might be, the sheer volume of computation is responsible for the non real-time operation of the VGW package. The Suggestions for Improvement presented in this section also include other areas which came to light while writing this report. All suggestions, however, are aimed at making the VGW package run in real-time.

MODEL UPDATES

Updating the "Mathematical Model of an Air Cushion Vehicle" is a priority task. This task will resolve differences in the modelling by Cummings, et al, and data which has been implemented at NTEC. The simulation program which was delivered and is described in this report reflects the model as given and does not necessarily represent, in all areas of modelling, the true vehicle. The mathematical model was developed from Bell's preliminary design. Now that JEFF-B has been built, full-scale actual data should exist to properly update the model.

Some of the areas to be investigated are as follows:

air flow rates
air pressures
skirt gap areas - skirt clearances
natural frequency of vehicle
drag coefficients
engine characteristics
wind forces
vehicle weights and inertias
fan characteristics

The model represents the best estimate from Bell's engineering experience and now real data should be available to update the pilot trainer to represent the actual vehicle.

CLEANUP AND TIMING IMPROVEMENTS

Simulation program cleanup includes tasks as diverse as coding and modelling errors which have been identified at both CSDL and NAVTRAEQUIPCEN. Although, these have been properly implemented at NTEC, a proper program update should be accomplished so that program listings and documentation can remain current at CSDL and NAVTRAEQUIPCEN.

There are several areas where execution Timing Improvement can be made. Perhaps other areas exist, but it would necessitate a close look by an experienced assembly language programmer. The Timing Improvements described here are exclusive of the VGW package, which is described in the next section.

A time savings can be realized by changing the model of the gas turbines Combining the three port/starboard turbines into one engine will save considerable computation. This, of course, is not realistic but there is only one pilot control each for port and starboard simulated

A small execution time savings can be realized by using a constant wind speed and set per run. This would necessitate changing these parameters during initialization only. The execution time saved would be that of computing the sine and cosine functions.

The seaway simulation timing can be improved by performing algebraic manipulation of the equations and by using the piecewise approximation to the cosine function.

There is a considerable potential of execution time savings by improving the cushion pressure solution of the 6 simultaneous equations. The present method uses a Newton-Raphson technique to compute a 6 x 6 matrix and solution by matrix inversion. To create the matrix, many computations are involved and 32 function calls to 7ASQRT and 7DASQRT per iteration. Although this is an excellent method in accuracy and convergence, another method may take many more iterations, but require less computational time.

A completely different method for solving for the cushion pressures would completely relieve the execution duty cycle by implementing the 6 simultaneous equations on the analog computer. This would be a fairly simple implementation in the hybrid facility at NTEC and should guarantee convergence.

The last method that could be implemented would be the complete programming in fixed point arithmetic of the all digital simulation. This would be a last resort item after all other possibilities have been explored. There is great flexibility now by using floating point arithmetic, and fixed point arithmetic would limit that flexibility. Also, this would necessitate a nearly complete redevelopment of the real-time simulation program.

VGW TIMING

The Vehicle Generated Wave (VGW) model presents the most significant problem to real-time operation. The huge number of calculations to be performed presents the greatest difficulty. At the present time, a complete execution to calculate 25 VGW heights takes approximately 1.5 seconds. For each wave height, 82 kernel values must be calculated by table lookup and interpolated.

Several possibilities exist to decrease the execution time. The first and foremost must be to eliminate the linear interpolation to obtain the kernal value. This would necessitate taking the kernel table that now exists and updating it with constant intervals and by minimizing kernel error.

Since the kernel values are already fixed point in the table, the floating conversion could be eliminated for the 82 kernel values per point and then the convolution integral could be calculated in fixed point arithmetic. By doing this, there would be only a total of 25 floating point conversions instead of 25 x 82 (2050).

After all modelling and timing updates have been performed, a final determination needs to be made as to whether the VGW model can be run in real-time. The object is to remove the first order lag on VGW heights (subroutine: FILTER). By this time, there may be sufficient execution time remaining in the duty cycle to include the VGW calculations. Perhaps not all 25 points, but 5 points a cycle would probably be fast enough for true vehicle updates. This would yield a total update rate of the VGW heights of 250 msec (4HZ). Studies would then have to be made to determine the adequacy of the update rate.

The final method of solution would be to off-load all VGW height calculations to a fast mini-computer for execution. Then the only load on the simulation program duty cycle would be the data transfer between computers, which would not be trivial.

PILOT VISUAL AFFECTS AND MOTION SIMULATIONS

For a pilot trainer to be truly effective, proper visual affects should be simulated for the pilot's awareness. Affixing the pilot station to a motion platform is also an added affect on the pilot's awareness. Concurrent development of the hardware/software interface may be of benefit. At the very least, program listings and documentation would be jointly held and current

REPORT UPDATES

In no case should changes be made to the real-time simulation of JEFF-B without adequate documentation. This documentation should not consist of merely handwritten notes or markups within this report. This report should be properly updated at proper intervals during simulation development.

DISTRIBUTION LIST

Naval Training Equipment Center Orlando, Florida 32813	23
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Commander, Naval Sea Systems Command Naval Sea Systems Command Headquarters Washington, DC 20362 ATTN: James Sejd, Code SEV 03724	4
Commander Naval Ship Research and Development Center Bethesda, Maryland 20034 ATTN: George Wachnick, Code 1182	3
Commanding Officer Naval Coastal Systems Laboratory Panama City, Florida 32401	1
Chief of Naval Education & Training Support Pensacola, Florida 32508 ATTN: Code N-2	1
Bell Aerospace Company P.O. Box 29307 New Orleans, LA 70129	1
Scientific Technical Information Office NASA Washington, DC 20546	1
Officer in Charge MDWP Contract Supv Staff Saclant ASW Research Center APO New York 09019	1
NASA Langley Resch Ctr Library Langley AFB, VA 23365	1

DISTRIBUTION LIST CONTD

Chief of Naval Training Code 017 NAS Pensacola, FL 32508	1
Commander Naval Weapons Lab Attn: Tech Library Dahlgren, VA 22448	1